AL-TP-1991-0016

ARMSTRONG LABORATORY

INTEGRATED MAINTENANCE INFORMATION SYSTEM DIAGNOSTIC MODULE (IMIS-DM) Version 5.0

Garth Cooke
Nicola Malorana
Theodore Myers
Johnnie Jernigan

Systems Exploration, Incorporated
5200 Springfield Pike, Suite 312
Dayton, OH 45431

Eric N. Carlson, 2Lt, USAF

HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION
Wright-Patterson Air Force Base, OH 45433-6503

June 1991

Final Technical Paper for Period February 1990 – January 1991

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5000

NOTICES

This technical paper is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

ERIC N. CARLSON, 2Lt, USAF
Contract Monitor

BERTRAM W. CREAM, Technical Director
Logistics Research Division

JAMES C. CLARK, Colonel, USAF
Chief, Logistics Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1991	Final Technical Paper - Feb 1990 - Jan 1991

4. TITLE AND SUBTITLE
Integrated Maintenance Information System Diagnostic Module (IMIS-DM) Version 5.0

5. FUNDING NUMBERS
C - F33615-88-C-0004
PE - 63106F
PR - 2950
TA - 00
WU - 21

6. AUTHOR(S)
Garth Cooke
Nicola Maiorana
Theodore Myers
Johnnie Jernigan
Eric N. Carlson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systems Exploration, Incorporated
5200 Springfield Pike, Suite 312
Dayton, OH 45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
Wright-Patterson Air Force Base, OH 45433-6503

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL-TP-1991-0016

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Air Force Human Resources Laboratory/Combat Logistics Branch (AFHRL/LRC) is engaged in the research and development (R&D) of an Integrated Maintenance Information System (IMIS). This system will be capable of accessing and integrating information from several Air Force data bases to provide technical support to the maintenance technician. This support will be provided by a portable computer maintenance aid which will provide instructions for accomplishing maintenance tasks. A diagnostic module will be contained in the portable computer software to help the technician in performing complex diagnostic tasks. This paper is a comprehensive description of the IMIS Diagnostic Module (IMIS-DM) Version 5.0.

14. SUBJECT TERMS			15. NUMBER OF PAGES
diagnostic modeling diagnostics fault isolation	IMIS IMIS-DM maintenance technician	troubleshooting	58 16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

LIST OF FIGURES

LIST OF TABLES

PREFACE

This paper details the IMIS-DM designed for the AFHRL, Combat Logistics Branch (LRC), under the terms of contract #F33615-88-C-0004, Task Order #0018. The task monitor and principal investigator for the Air Force Human Resources Laboratory was Lt. Eric Carlson, AFHRL/LRC.

Research was performed by the Dayton regional office of SEI. Principal investigators were Garth Cooke, Nicola Maiorana, Theodore Myers, and Johnnie Jernigan.

SUMMARY

This technical paper summarizes the work performed by Systems Exploration, Inc. (SEI) to redesign and enhance the Integrated Maintenance Information System Diagnostic Module (IMIS-DM) and diagnostic maintenance environment. The IMIS-DM is part of an ongoing research and development (R&D) effort by the Air Force Human Resources Laboratory (AFHRL) to access and integrate maintenance information from multiple sources and present the information to technicians through a rugged, hand-held computer. The diagnostic module provides technical support to the maintenance technician by furnishing a wide range of capabilities to assist in the selection of an efficient sequence of maintenance tasks.

R&D in diagnostics has led to an IMIS-DM which provides a wide range of capabilities that assist the technician in selecting an efficient sequence of maintenance tasks. These tasks lead to rapid and effective repair of failed components. The module was designed to work efficiently in an "on-equipment" maintenance environment. The technician's job in this environment uses the module to isolate problems to a replaceable component level rather than to the lowest possible level at which a failure might occur. However, the module is equally effective at the lower levels with appropriate adjustments to the data base.

The IMIS-DM uses algorithms to identify the test and repair activity sequence most likely to result in a repaired system in the minimum amount of time. The algorithms calculate the likelihood of component failures and task accomplishment times to recommend the next sequenced action. The module determines these dynamically at each stage of the diagnostic session rather than exhaustively precalculating them to establish a fixed-sequence decision tree. Finally, the algorithms provide the technicians with lists of available actions which might prove effective in repairing the system. The lists are rank-ordered by calculated probability of success. The highest probability action is recommended; however, the technician is free to choose among the available alternatives. Once the technician completes an action, the next recommended action is then calculated based upon the results of the previous action.

This paper provides complete documentation of Version 5.0 of IMIS-DM implemented in Smalltalk/V and contains enhancements not included in earlier documentation (AFHRL-TP-90-13 and AFHRL-90-53). Enhancements in this version include (a) a technical data navigation function designed to assist in the selection of appropriate technical data sequences needed for such complex activities as cannibalization of parts from another aircraft, (b) decision and troubleshooting assistance for Can Not Duplicate (CND) malfunction occurrences, and (c) an Estimated Time in Commission (ETIC) estimator. In addition, the multiple fault probability calculation was modified to provide a faster response in the event of large implicated fault groups.

SUMMARY

This technical paper summarizes the work performed by Systems Exploration, Inc. (SEI) to redesign and enhance the Integrated Maintenance Information System Diagnostic Module (IMIS-DM) and diagnostic maintenance environment. The IMIS-DM is part of an ongoing research and development (R&D) effort by the Air Force Human Resources Laboratory (AFHRL) to access and integrate maintenance information from multiple sources and present the information to technicians through a rugged, hand-held computer. The diagnostic module provides technical support to the maintenance technician by furnishing a wide range of capabilities to assist in the selection of an efficient sequence of maintenance tasks.

R&D in diagnostics has led to an IMIS-DM which provides a wide range of capabilities that assist the technician in selecting an efficient sequence of maintenance tasks. These tasks lead to rapid and effective repair of failed components. The module was designed to work efficiently in an "on-equipment" maintenance environment. The technician's job in this environment uses the module to isolate problems to a replaceable component level rather than to the lowest possible level at which a failure might occur. However, the module is equally effective at the lower levels with appropriate adjustments to the data base.

The IMIS-DM uses algorithms to identify the test and repair activity sequence most likely to result in a repaired system in the minimum amount of time. The algorithms calculate the likelihood of component failures and task accomplishment times to recommend the next sequenced action. The module determines these dynamically at each stage of the diagnostic session rather than exhaustively precalculating them to establish a fixed-sequence decision tree. Finally, the algorithms provide the technicians with lists of available actions which might prove effective in repairing the system. The lists are rank-ordered by calculated probability of success. The highest probability action is recommended; however, the technician is free to choose among the available alternatives. Once the technician completes an action, the next recommended action is then calculated based upon the results of the previous action.

This paper provides complete documentation of Version 5.0 of IMIS-DM implemented in Smalltalk/V and contains enhancements not included in earlier documentation (AFHRL-TP-90-13 and AFHRL-90-53). Enhancements in this version include (a) a technical data navigation function designed to assist in the selection of appropriate technical data sequences needed for such complex activities as cannibalization of parts from another aircraft, (b) decision and troubleshooting assistance for Can Not Duplicate (CND) malfunction occurrences, and (c) an Estimated Time in Commission (ETIC) estimator. In addition, the multiple fault probability calculation was modified to provide a faster response in the event of large implicated fault groups.

I. INTRODUCTION

This paper completely describes the current Integrated Maintenance Information System Diagnostic Module (IMIS-DM). The Air Force Human Resources Laboratory (AFHRL), Human Systems Division (HSD), Air Force Systems Command (AFSC) is developing a prototype IMIS. The IMIS prototype will demonstrate the capability to access and integrate maintenance information from multiple sources and present the information to technicians through a rugged, hand-held computer. Results of the program will form the basis of requirement specifications for such a system. AFHRL is performing preliminary research in many areas key to the success of the IMIS concept. Specifications developed by the project will be proven effective through rigorous field evaluations by Air Force maintenance technicians. As a result, IMIS will improve the capabilities of maintenance organizations to efficiently use available manpower and resources, and effectively meet combat sortie-generation requirements.

The modern maintenance environment is becoming increasingly inundated with additional computer-based information systems. Examples include the Comprehensive Engine Management System (CEMS), the Core Automated Maintenance System (CAMS), and the Automated Technical Order System (ATOS). Each new maintenance aid forces technicians to learn yet another system. AFHRL is developing the IMIS to facilitate the use of the valuable information these new systems offer, while eliminating the specialization required for each. The IMIS will use a very small, portable computer/display, referred to as the Portable Maintenance Aid (PMA), to interface with on-aircraft systems and ground-based computer systems to provide a single, integrated source of the information needed to perform maintenance on the flight line and in the shop. The IMIS will consist of a workstation for use in the shop, a portable computer for flight line use, and an aircraft interface panel for interacting with aircraft systems. The IMIS will access, integrate, and display maintenance information for use by the technician. It will provide the technician direct access to several maintenance information systems and data bases including historical data collection and analysis, supply, technical orders (TOs), and automated training systems. The IMIS will display graphic technical instructions, provide intelligent diagnostic and rectification advice, provide aircraft battle damage assessment aids, analyze in-flight performance and failure data, analyze aircraft historical data, and interrogate on-aircraft built-in-test capabilities. It will also provide the technician with easy, efficient methods to receive work orders, report maintenance actions, order parts from supply, and complete computer-aided training lessons. The PMA will function independently to display all the information the technician needs for on-equipment maintenance. Even if the base-level computer systems are unavailable or the aircraft systems are malfunctioning, the IMIS will be able to display TO information and the diagnostic aids to the technician. The PMA will make it possible to present quality information by taking advantage of the computer's ability to interact with and tailor information to technicians of varying levels of expertise.

Based on previous maintenance evaluations and scenarios, IMIS was divided into four major subsystems: (a) the technician's PMA; (b) an aircraft maintenance panel connected to on-board computers and sensors; (c) a maintenance workstation connected to various ground-based computer systems; and (d) sophisticated integration software that will combine information from multiple sources and present the data in a consistent way to the technician. The technician's primary interface with the IMIS will be the PMA--a compact, lightweight, battery-powered portable computer rugged enough for flight line use. A library of removable memory cartridges will store all TO information and diagnostic aids needed for a single weapon system. The memory cartridges will be designed for fast and easy updating. A digital radio link will be capable of transmitting and receiving both voice inputs and binary data. Advanced digital transmission techniques will allow multiple users on the same frequency, thereby reducing the radio frequency clutter on the flight line. A high-resolution, flat-panel display will display data clearly under all lighting conditions. The human-computer interface will be designed for ease of operation to eliminate the need for the user to have typing skills. The PMA will have sufficient processing power to quickly display complex graphics and provide rapid responses to the technicians's

requests. Interactive troubleshooting routines and artificial intelligence-based diagnostic aids will provide advice for difficult fault-isolation problems.

The technician will be able to accomplish most aircraft maintenance tasks without climbing into the cockpit. An aircraft maintenance panel on the outside of the aircraft will provide the interface to on-aircraft systems. The maintenance panel will allow the technician to interact with aircraft systems easily and will reduce the need to climb into the cockpit. This panel will consist of a control and display unit and an interface connector for the IMIS PMA. The aircraft maintenance panel will be used to retrieve data on configuration and subsystem status, interrogate built-in-test and on-board diagnostics, and upload and download mission software. The panel may also be used in conjunction with the PMA for extended diagnostics and troubleshooting.

The technician will interface with ground-based systems through a maintenance workstation. The desktop workstation will include a full keyboard and an interface computer. The interface computer will have the protocol software required to access the other available data systems. The PMA will connect to the workstation. The workstation will provide the technician with the capability to access and exchange information with systems such as CAMS and ATOS.

The most beneficial feature of the IMIS for the technician will be the integration of information. Instead of dealing with several automated systems and accessing separate groups of information through several devices, the technician will access all information through a single device. At a superficial level, the system will integrate information by employing standard commands and display formats. At a deeper level, through sophisticated software, the system will integrate information from all available sources to provide a coordinated maintenance package.

The IMIS-DM is a software application module in the technician's PMA that helps the maintenance technician isolate and repair faulty aircraft components. The IMIS-DM's key features are designed to minimize repair time rather than fault isolation time. This philosophy takes advantage of instances when a rectification would have a higher probability of repairing a problem faster than isolating the problem with tests and then repairing the problem. The IMIS-DM has special subroutines that perform symptom/component matching, taking into account component histories, probabilistic data, logistic constraints, and operational constraints.

II. IMIS-DM THEORY

We compared three baseline modeling techniques in developing the IMIS-DM diagnostic aiding strategy. This comparison considered system history and design knowledge. The modeling techniques consist of fault modeling versus component modeling, fault isolation versus fault rectification, and information gained versus cost expended. Although each technique was evaluated as an independent approach, findings proved that combining beneficial attributes of related techniques was more effective. The following discussions provide descriptions of the comparisons and the combined attributes selected to develop effective diagnostic modeling techniques. The results accurately set up and attack the problem at hand, maximize the information gained, and minimize the cost expended.

Fault Modeling Versus Component Modeling

A component modeling technique maps each test result, fault code, or symptom to a plausible set of components. Rectification actions are then considered as a maintenance technician's action upon a single component. In flightline aircraft maintenance, problem rectification is frequently limited to "box swapping" or swapping of Line Replaceable Units (LRUs). As a result, repair of broken or malfunctioning components as the goal of the diagnostic exercise is an accurate model. In addition, if the end item (the LRU as component) is disassembled to the Shop

Replacement Unit (SRU) level, then modeling to the SRU as the component of interest would be an accurate model of a lower level of maintenance. A model based on this technique quickly becomes intractable due to large numbers of special cases and seemingly complicating information.

Fault modeling is an improvement to the component modeling. Since most components are assemblies of lower level parts, failures of different parts in the component may have different effects. Any of these effects may indicate a malfunction of the component. Hence, when one defines a fault as the manifestation at the component level of a subcomponent's failure, then a component can be said to contain one or more potential faults. All of these faults can be readily defined through engineering analysis. The advantage of this technique is that faults are discrete, observable, or measurable while failures may be hidden; and the faults identifications are more descriptive than "malfunctioning component."

The objective of the diagnostic effort then is to isolate a fault rather than to isolate a faulty component. This fault modeling scheme greatly improves program effectiveness and tractability. The fault modeling scheme also provides significant amounts of failure data that may prove very valuable in subsequent maintenance activities at the SRU level. The problem with this technique is that the flightline maintenance technician swaps components rather than faults. This is a serious drawback because a model with a high level of fidelity to maintenance practice is essential to ready acceptance.

Therefore, we combined the two techniques. This solution was achieved by considering each component as "a bucket of potential faults." Each fault can be mapped to a rectification action. A rectification action may be the replacement of an LRU or some other maintenance action such as an adjustment, alignment, or a temporary repair. The set of rectification actions then maps to a single component upon which the rectification actions occur. After formulating and combining these two techniques, we produced a reachability matrix that mapped the diagnostic parameters of the system under investigation (see Table 1).

Table 1. Fault - Rectification - Component - Test Mapping

		MOT								
	S0	**T0**	**T1**	**T2**	**T3**	**T4**	**R0**	**R1**	**R2**	**R3**
F0	1	1	1	0	0	F	1			
F1	1	1	0	0	1.1	C		1		1
F2	1	0	0	1	1.2	K			1	1
COMPONENT							**A**	**B**	**B**	**B**

F-Fault S-Symptom R-Rectification T-Test
MOT-Multiple Outcome Test FCK-Functional Check

In this matrix, the symptom (SO) implicates the faults (F0, F1, and F2) as potential causes of the observed problem. Each of the tests (T0, T1, etc.) is shown to span, or be affected by, one or more of the faults (shown as a 1). Test 3 is a Multiple Outcome Test (MOT) that has two discrete outcomes. It specifically measures for the presence of F1 and F2 in a single test procedure. Rectifications (R1 and R2) are maintenance actions that do not require removal or replacement of Component B. R3 is an action that requires removal and replacement of Component B.

Fault Isolation Versus Fault Rectification

The maintenance technician is frequently faced with the diagnostic problem of having two or more components in a system under investigation with no tests available to determine which of

the components is faulty. The goal is to fix the system by replacing the components most likely to contain the failure and to minimize system downtime.

The initial assumption in developing the diagnostic module was that the technician would always attempt to isolate the faulty component (fault isolation) through available tests before attempting any repair action. Two factors led to this conclusion. First, fault isolation conserves supplies. Any attempt to repair prior to fault isolation can lead to the replacement of a component that is not faulty and needlessly depletes units from supply. Second, fault isolation conserves manpower by eliminating the effort required to remove and replace components that are not faulty.

This fault isolation strategy had to be reexamined. Given a particular symptom or set of symptoms, the set of possible faults may include a subset from one component that is so likely to have caused the symptom that an immediate rectification action is warranted. This sort of alternative may be particularly attractive when the system is badly needed for operational requirements. Under pressing time constraints, even when tests are available for fault isolation, analysis should provide a series of recommendations to repair a system in minimum time. However, if test times approach or become large compared to replacement times, the analysis might yield a swap-first decision with a decreasing probability that the swap action will fix the fault. Such a situation could be very inefficient when there are no pressing time considerations or there are few spare components. To solve this problem, the Second Step probability of success was developed. This method provides an examination of what the maintenance technician could expect to find at the end of the second upcoming maintenance event in the diagnostic sequence.

Information Gained Versus Cost Expended

Initial development of the diagnostic algorithms and analyses focused on evaluating available options based on the information gained from the test results. This approach minimizes diagnostic steps in fault isolation. For example, several tests are frequently available in a diagnostics session to further the process. The task facing the maintenance technician is to select the most efficient test available. However, the information from a binary test can be a passing result and a failing result. The best test, therefore, is one which maximizes the information gained from whichever result occurs. We maximized the information gained by combining a split-half strategy with failure rates (FRs) of plausible faults.

I_j = The information value gained from performing test j
FR_i = The failure rate of the ith fault = 1/Mean Time Between Failures $(MTBF)_i$
$FR(1)$ = The failure rate of a plausible fault spanned by test j
$FR(0)$ = The failure rate of a plausible fault not spanned by test j
FR_{PS} = The failure rate of the plausible set = $\sum FR_i$

$$I_j = \frac{\sum_{i=1}^{PS} FR(1)}{FR_{PS}} \times \frac{\sum_{i=1}^{PS} FR(0)}{FR_{PS}} \qquad (1)$$

This strategy provided a means for selecting tests based on information gained but did not fully justify performing a time-consuming test. Other available tests may not provide as much information but may require a fraction of the time to complete. Certainly, time to accomplish a test should be considered a cost metric associated with that test. Excessive costs can accrue from an

information gain strategy that maximizes the information gained but provides little insight into the cost of obtaining that information. This observation led to the development of the analyses that evaluate tests by calculating information gained per unit of time invested.

III. IMIS-DM DESIGN

IMIS-DM Controller

The IMIS-DM controller is an executive system that controls and manipulates three subsystems: (a) an applications system (such as IMIS-DM, pre/post flight, phase inspection, and weapons load), (b) the data base module, and (c) the presentation system. Figure 1 illustrates the system. The applications system is identified as the diagnostic module.

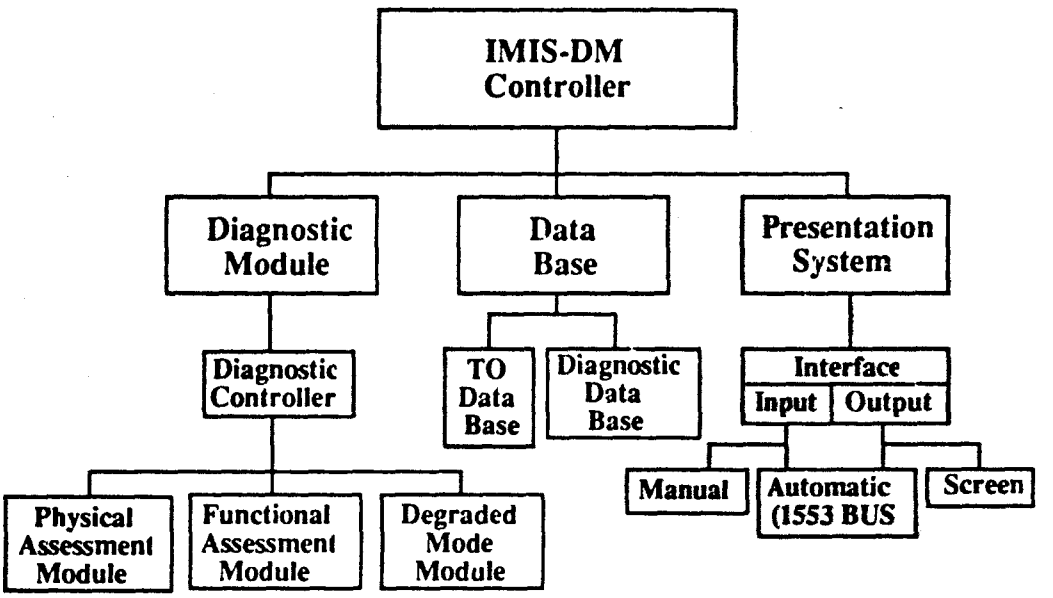


Figure 1. IMIS-DM Controller System.

The diagnostic module comprises four major submodules: (a) the diagnostic controller, (b) the physical assessment module, (c) the functional assessment module, and (d) the degraded mode module. The diagnostic controller module manages data items, diagnostic groupings, performance of its submodules, and interfacing to the IMIS-DM controller. Interfaces between the IMIS-DM controller and the diagnostic controller provide the means to extract diagnostic data from the data base and present information to the presentation system. The assessment modules are the subject of this technical paper and are described in detail in subsequent sections.

The data base module, in Content Data Model (CDM) hierarchical format, contains both TO and diagnostic data information.[1] The TO information consists of procedural text, graphic

[1]The CDM is a Standard Generalized Markup Language (SGML) Document Type Definition (DTD) that describes the logical structure for a data base of technical information. The model is under development by the Air Force Human Resources Laboratory/Logistics Systems Branch (AFHRL/LRC) at Wright-Patterson AFB.

illustrations, and data element information. The diagnostic data contain mapping and probabilistic data for faults, tests, symptoms, and rectifications. Diagnostic data are used by the IMIS-DM; the TO information is used mainly by the presentation system.

The presentation system provides interfaces for aircraft system health checks, maintenance technician input, display of diagnostic module results, technical data for maintenance tasks, and system specific graphics. System health checks use data from Military-Standard-1553 (MIL-STD-1553) data bus downloads and input from the maintenance technician to give the diagnostic module pertinent information about the aircraft under investigation, logistic constraints, operational constraints, and task performance. The diagnostic module gives the presentation system information for fault isolation and repair information. The data base module supports both the diagnostic module and the presentation system.

Smalltalk/V Module Development

The first diagnostic module was written using the C programming language. We converted this module from C to Smalltalk/V. Smalltalk is a high-level, object-oriented (OO) programming environment. There are two reasons why Smalltalk was selected as a development language. The first is the rapid prototyping capabilities available through the Smalltalk environment. This environment contains most of the low-level functions used in software development. The environment also allows a programmer to compile and test smaller pieces of code within the environment. Moreover, the Smalltalk environment allows the code to be reused. The second reason for selecting Smalltalk is for ease of integration into the presentation system.

Diagnostic Module Controller

The diagnostic module controller uses hierarchical data from the CDM data base. The controller creates multiple diagnostic groups based on interdependent symptoms. These groupings are based on faults relating to common tasks (tests or rectifications).[2] Each diagnostic group is independent, and the status of the faults (plausible, exculpated, rectified, and so on) is stored in the diagnostic group to which it belongs.

To begin the diagnostic process, symptoms are passed to the diagnostic module controller by the IMIS-DM controller. Each symptom maps to, or spans, a set of faults. Furthermore, these spanned faults can map to a set of subfaults, forming a fault tree. The lowest level faults in the fault tree are evaluated to find tests and rectifications pertaining to the original symptom. Diagnostic groups are then created by categorizing symptoms with common tests and rectifications. Symptoms that do not possess common tests and rectifications are considered independent and are categorized into separate diagnostic groups.

[2]The term "fault" is used to describe a functional or physical manifestation of some low-level physical failure. Throughout this report, we use the term fault to refer to both the single fault present in the system (that which is bad) and to all possible faults that can cause a symptom. A symptom is a machine-generated code or a verbal description indicating a malfunction exists within a system. The symptom implicates one or more possible faults which can cause the malfunction. Use of these terms can be confusing in a hierarchically arranged data base such as the CDM, because the thing called a fault at one level of the hierarchy may be referred to as a symptom in a lower level of the hierarchy.

Physical Associations Model

The diagnostic module, originally designed to evaluate fault isolation and repair alternatives from almost a purely functional standpoint, has been enhanced to perform both functional and physical assessment. When diagnostics are approached from a purely functional standpoint, we cannot adequately address events causing malfunctions of other components, or malfunctions caused by a nearby physical event. For example, a technician may enter a compartment of an aircraft and observe that hydraulic fluid has leaked all over the bay, causing a failure in an LRU. Repair of that LRU would not be appropriate until the hydraulic line is repaired and the bay is cleaned. Many external causes of the functional problem (i.e., aircraft battle damage, bird strikes, environment extremes) could create problems with the system under investigation. Hence, some physical model should be developed to produce an efficient cause and effect or physical association isolation and repair strategy.

To work with physical associations, we need to look at what makes a physical association reasonable. One key element is physical proximity. However, proximity is not enough by itself. There must be a physical event occurring that can affect systems, components, or parts near the event for a valid physical association. A physical event in this context implies that some foreign agent can act externally to the affected component and cause a failure. This implication, then, implies that there must be a source of the foreign agent, and the component(s) in the vicinity must be vulnerable to damage by that agent.

If we only consider the battle damage source, there are many assessment models available that may prove more effective for this limited role than this modified diagnostic module. However, if we consider other sources of damage, such as heat, damaging liquid contamination, and high vibration levels, we can look inside the weapon system to find potential sources of the damaging physical agents. When we look at the weapon system for these sources, we define the limits of the universe of possible physical associations. Most normal physical hazards associated with operating in an airborne environment are already built into the MTBF and the fault weightings considered in the functional diagnostic module (e.g., routine g-loads, normal vibration levels, operating temperature extremes, humidity, and so on). Consequently, the physical associations model must address hazards outside the range of "normal" hazards associated with operating in an airborne environment.

Therefore, the hazards to which aircraft parts and components may be subjected are few. Among these are (a) temperature extremes, primarily high temperatures; (b) liquids such as fuel, lubricants, hydraulic fluid, water, and so on; and (c) physical abuse. Physical abuse is the most widespread category because it includes both internal and external sources and has a wide range of potentially severe effects. These sources can be internal (explosion of Cartridge Activated Device (CAD); rupture of pressure vessels; slow burning/misfire of CAD; and loss of containment of high energy, spinning devices) or external (dropped objects, bird strike, mid-air collision, Foreign Object Damage (FOD), and battle damage).

Physical Effects Mapping

In a restricted hazards list we must identify, within some boundary (e.g., an avionics bay, an engine bay), each of the components containing hazards to either itself or to other components within the boundary. Finally, we must identify those components within the boundary that are vulnerable to these hazards. Vulnerable, in this case, refers to the functional model and implies some component may not operate within prescribed functional limits because of the effect created by a hazard normally contained within some other component.

The relationship where components are vulnerable to hazards contained within some other component is referred to as a migration of the hazard. Each hazard in the restricted hazards list has individual migration traits and must be mapped accordingly to the particular hazard. For instance, high-temperature hazards tend to manifest themselves where the heat is exhausted and in the upper area of a bay, whereas fluids tend to migrate with gravity to the lower areas of the bay. Physical abuses also tend to follow the migration concept and tend to affect their components within restricted geometric bounds. The logical approach to tracking and marking boundaries or hazard areas can be described within a three-dimensional coordinate system.

Assuming data to support a migration of hazard problem are available, then the diagnostic model must be altered to consider the effects of these physical relationships. If, during a diagnostic or other maintenance task, evidence of the presence of a physical hazard is discovered, the maintenance technician is faced with two problems. First, he must identify the source of the hazard and, if necessary, rectify the failure that released the hazard.[3] Then, he must identify and, if necessary, repair any failed components. The data to support this scheme could be represented as in Table 2.

Physical Model Operation

The logic flow for physical assessment model appears as in Figures 2 and 3. This logic flow maximizes the capabilities built into the current functional assessment module and expands upon the processing and modeling schema. The branching and control mechanisms have been built into the current diagnostic module using the logic-based PROLOG of Smalltalk/V. This section briefly describes the physical assessment module's operation. Details of the algorithms and operations in these figures can be found in Cooke et al. (1990b).

Upon initialization, manual and automatic system health information and physical evidence information (e.g., hazard codes, location) are entered. The diagnostic controller module then determines how to approach the diagnostic problem based on the data entries observed and diagnostic groupings available. The functional assessment approach is initiated when only functional symptoms have been observed and there is no physical evidence of a potential problem. If physical evidence appears during functional isolation and repair assessment, the diagnostic controller module redirects efforts to physical isolation and repair assessment without losing the information gained from previous actions.

Physical assessment can also be initiated when physical evidence of a problem is observed prior to normal diagnostics, e.g., bullet hole in engine bay, hydraulic fluid in avionics bay. The physical assessment module performs diagnostics by first addressing source faults. Figure 3 provides the logic flow for alleviating source faults (A5 on Figure 2). Initially, source repair is performed on components showing obvious physical damage, then the functional assessment module is activated to alleviate source faults that are not so obvious. The second step in the physical assessment module's logic is to isolate and repair affected components. Again, the approach to physical assessment of affected components is to first repair obviously damaged components. When all known damaged components are repaired, the diagnostic controller module reverts to functional assessment to complete the aircraft repairs.

[3]The individual technician is central to the discussions in this paper. For simplicity, we have used the singular pronoun "he" to designate the individual technician (whether that person is a man or a woman).

Table 2. Hazard Source and Effects Mapping

Rectification (Rect)	Hazard (H)	Vulnerability (V)	Source Fault(s) F(s)	Effect Fault(s) F(e)	Location (LOC)
A	a	-	1	-	
	b	-	2	-	
	-	c	-	1,3,4	
B	c	-	5	-	
	-	a	-	6,7,8	
C	-	c,a	-	9	
Rect ID	Hazard contained in Rect	Hazards Rect is vulnerable to	Faults which can lead to Hazard release	Functional faults which may result from exposure to Hazard	X,Y,Z Location for Rect

As shown in the figures, the physical assessment module's logic flow maximizes the maintenance technician's ability to evaluate and act upon physical evidence without being delayed by the details of the general technical data required to identify, clean up, and evaluate hazard exposure. However, the processing is available to provide additional information to assist the novice through the details if necessary.

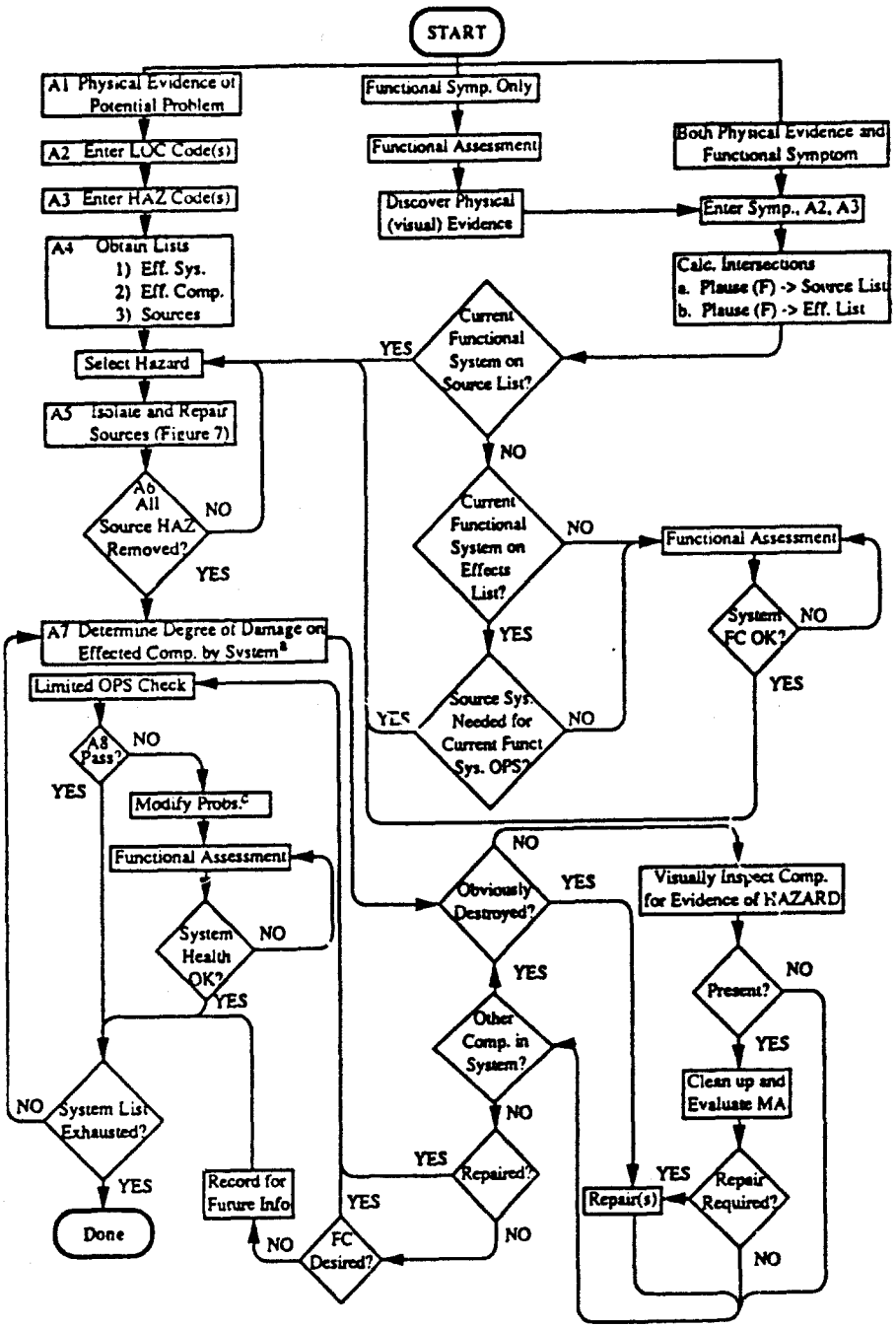
IV. IMIS-DM OPERATION AND ANALYSES

To develop the IMIS-DM, we developed and employed algorithms that incorporate the above techniques while handling specific constraints inherent in aircraft data and maintenance applications. The diagnostics module operates in three major subdivisions: (a) initialization, (b) fault manipulation, and (c) action ranking. During initialization, system descriptive data are loaded from a file system and specific constraint data are input through the computer keypad.

Faults are manipulated according to initial data entries and results of the technician's actions during the diagnostics session. Action ranking is performed recursively during the diagnostic session. It employs the analyses and calculations indicated by the current fault state. The current fault state is determined by the fault manipulation routines. This section explains the functionality and data processing for each of these activity subdivisions.

Logic Flow

Figure 4, Logic Flow, shows the sequencing of algorithms and analyses performed by the IMIS-DM. In the initialization process, the IMIS-DM accommodates both automatic and manual data input. Automatic data collection loads system specific data files from existing data bases and permits downloading of system health information from an aircraft data bus. The operator performs manual data entries such as symptoms, availability of parts and test equipment, critical states, and aircraft configuration. The diagnostic module then uses this information to evaluate fault combinations and to rank tests. Tests are then compared, by time analyses, to repair or replace activities, thus obtaining the highest likelihood of fixing the problem in the least time. Three lists of ranked tests and/or rectifications can be selected and presented to the maintenance technician: (a) ranked tests, (b) ranked rectifications, and (c) interleaved tests/rectifications. Although, a "best" action is recommended, the technician may select any of the listed options.



s = systems
c = components
Plause(F) = The set of plausible functional faults

Figure 2. Physical Assessment Model.

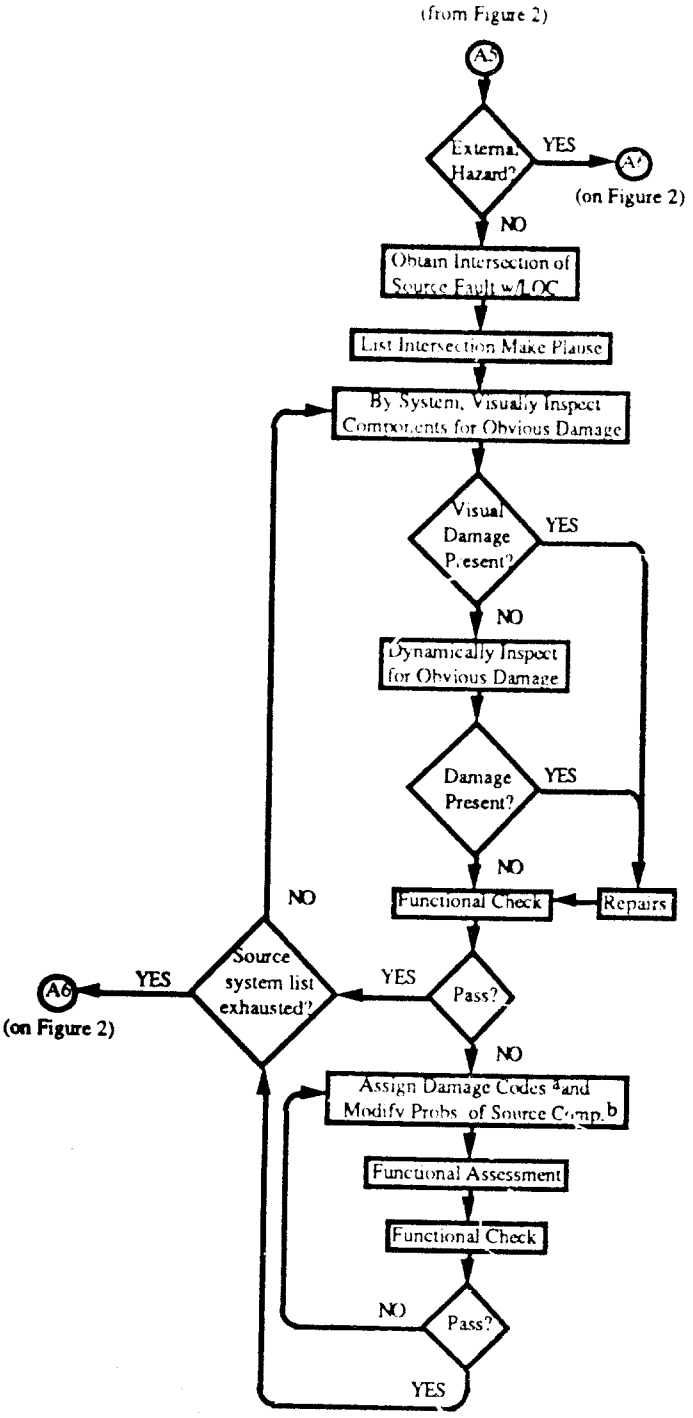


Figure 3. Isolate and Repair Sources.

Note.

[a]A description of the damage codes and formulas for modifying probabilities of components and faults is as follows:

Damage Code - (DC)

Code	Value	Desc
DS	100	Destroyed
DM	80	Damaged (dented, soaked, scorched)
SP	20	Suspected (dinged, scratched, dampened, liquid spots)
OK	1	No Apparent Effect

[b]Modifying probabilities of source components and faults. The following formulas are evaluated to obtain normalized modified source component and fault probabilities for a given location. Although Formulas 4 and 5 are only listed once, both component and fault probabilities are calculated using the same formulas but are performed separately.

$$Prob_{i,1} = \frac{\sum^{Comp} \frac{1}{MTBF\ (Fs)}}{\sum^{LOC} \frac{1}{MTBF\ (Fs)}} \qquad (2)$$

$$Prob_{i,1} = \frac{\frac{1}{MTBF(Fs)}}{\sum^{LOC} \frac{1}{MTBF(Fs)}} \qquad (3)$$

$$Prob_{i,2} = Prob_{i,1} * DC \qquad (4)$$

$$Prob_{i,F} = \frac{Prob_{i,2}}{\sum Prob_{i,2}} \qquad (5)$$

Where,

DC	=	component damage code value,
Fs	=	source fault,
Comp.	=	component,
LOC	=	location,
$Prob_{i,1}$	=	probability of the ith source component or fault,
$Prob_{i,2}$	=	modified probability of the ith source component or fault, and
$Prob_{i,F}$	=	normalized modified probability of the ith source component or fault.

[c]Modifying probabilities of affected components and faults. The following formulas are evaluated to obtain normalized modified effect component and fault probabilities for a given location. Although Formulas 8 and 9 are only listed once, both component and fault probabilities are calculated using the same formulas but are performed separately.

$$Prob_{i,1} = \frac{\sum_{Comp} \frac{1}{MTBF\,(Fe)}}{\sum_{LOC} \frac{1}{MTBF\,(Fe)}} \quad (6)$$

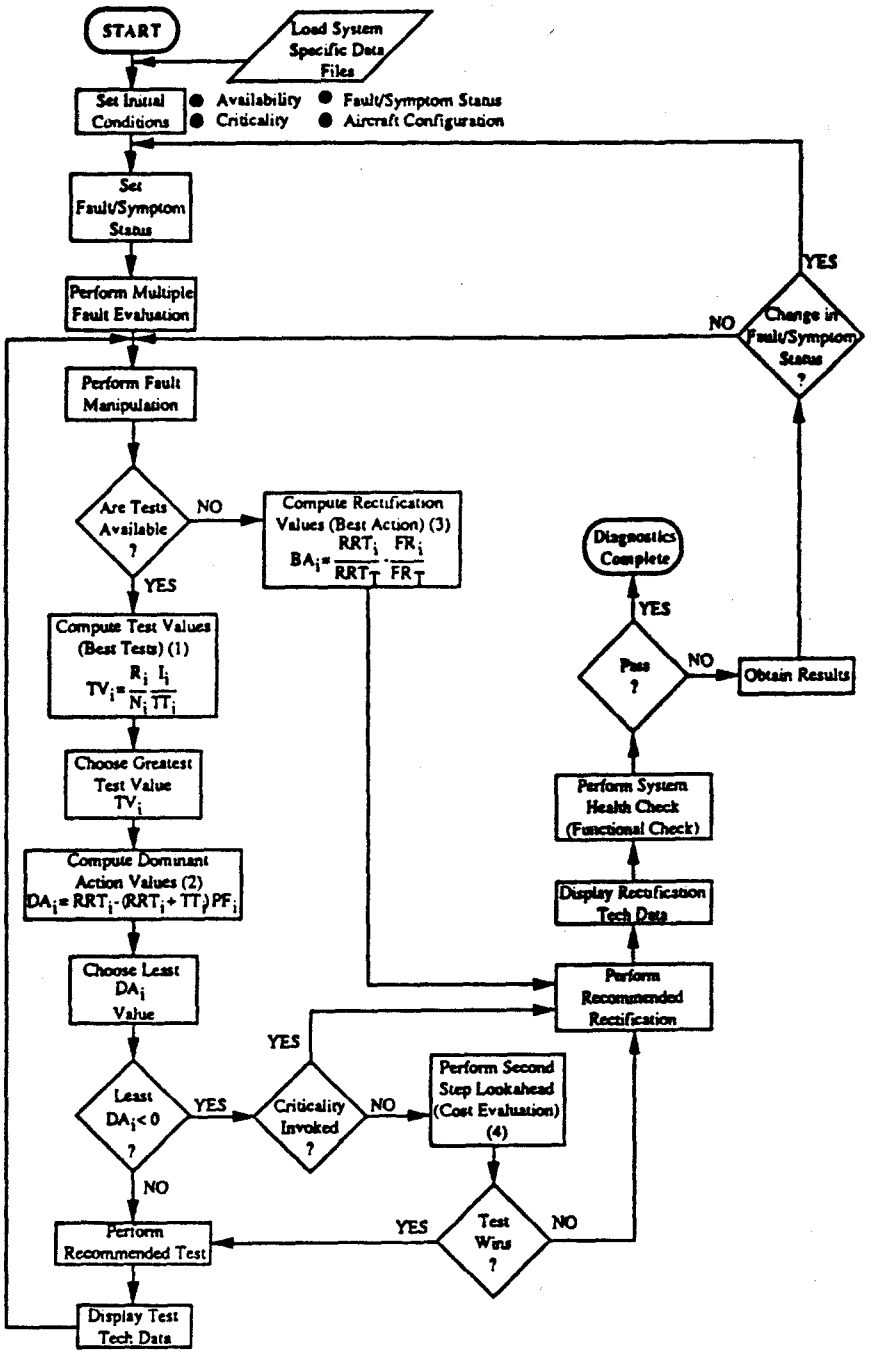
$$Prob_{i,1} = \frac{\frac{1}{MTBF(Fe)}}{\sum_{LOC} \frac{1}{MTBF(Fe)}} \quad (7)$$

$$Prob_{i,2} = Prob_{i,1} * DC \quad (8)$$

$$Prob_{i,F} = \frac{Prob_{i,2}}{\sum Prob_{i,2}} \quad (9)$$

Where,

Fe	=	affected fault,
$Prob_{i,1}$	=	probability of the ith affected component or fault,
$Prob_{i,2}$	=	modified probability of the ith affected component or fault, and
$Prob_{i,F}$	=	normalized modified probability of the ith affected component or fault.



Notes:
(1) Section IV. Page 19-21. Best Test/MOT.
(2) Section IV. Page 22. Dominant Action.
(3) Section IV. Page 23. Best Rectification.
(4) Section IV. Page 24. Second Step Look-Ahead.

Figure 4. Logic Flow.

When the technician selects a rectification or test, the presentation system displays TO instructions for performing the selected activity. If the selected action is a test, the diagnostic module performs fault manipulations based on the test outcome and repeats the evaluation of available options. If the selected action is a rectification or maintenance action, the IMIS-DM reinitializes the fault/symptom status using changes in the system health information obtained from a functional check. This procedure continues until the fault is isolated and the system is repaired.

Initialization

The initialization process provides the IMIS-DM with pertinent information about the aircraft system under investigation, dictates the sequence of events to follow to solve the problem, and is essential for diagnostic analysis performance. The information supplied to the diagnostic module during initialization includes up-to-date system fault/symptom and action parameters, current aircraft system health information, availability of required parts and test equipment, criticality of repairing that specific aircraft system, and configuration of the aircraft system under investigation. The following paragraphs describe these inputs and their functionality within the IMIS-DM.

Fault/Symptom Loading

To initialize the diagnostic module, system health information must be input from an outside source. System health information is entered as an observed malfunction of systems or machine-generated fault codes stemming from either automatic or operator-initiated Built-In Test (BIT). Possible symptoms and their associated potential faults are included in the system's data files. The observed symptom or fault code is the reason for starting a diagnostic session on a system.

Availability of Parts and Equipment

In many situations, part availability plays an important role in solving a maintenance problem. The IMIS-DM takes this factor into account. At some point in the diagnostic session, the recommended action may be to remove and replace a component. Furthermore, such a recommendation may occur before all the plausible faults are isolated to a single component. In such a case, it would be unwise to have the diagnostic module present a recommended action which could not be met because the necessary components were not available through the base support system. Consequently, at the start of the diagnostic session, the technician is given an opportunity to annotate any parts known to be unavailable. The diagnostic module adjusts for unavailable parts and avoids making remove and replace recommendations for these parts until they become available or until other actions have isolated all remaining faults to the unavailable part(s).

The concept of availability can be extended to include the equipment necessary to complete the diagnostics and resulting maintenance actions. The test equipment availability feature of the IMIS-DM takes into account the availability of test equipment and its direct effect on the ability to complete the recommended diagnostic tests. Consider a situation in which the IMIS-DM has selected a test as the best option; however, the equipment to perform that test is presently inoperative or unavailable. This makes the test a less-than-optimal choice since it cannot be readily accomplished. The IMIS-DM can consider test equipment availability when selecting the best option. This factor alleviates some frustration on the part of the technician faced with performing a test without the necessary equipment. Furthermore, this feature saves both time and money since the technician is warned against pursuing an action that cannot be performed. Once availability is set, the IMIS-DM finds the tests and components that are affected, if any, and marks them for future reference. If an inappropriate action is selected, the IMIS-DM displays the action as an invalid option.

Criticality

Criticality is a term used to designate some system functions essential for operational requirements. When referring to aircraft maintenance, the assignment of critical functions signifies that potential faults in those functions must be fixed or confirmed as good before the aircraft leaves for its next mission. As an example, assume that a weapon system has both air-to-air and air-to-ground capabilities. If the next scheduled mission is for an air-to-air combat capability, then air-to-air capabilities may be designated critical and air-to-ground capabilities noncritical.

During initialization, the technician has an opportunity to designate a function or group of functions critical. All potential faults contained in the components required to accomplish the critical function are then identified as critical. The diagnostic module then searches for plausible faults (implicated by symptoms) identified as critical. This group is designated the critical set of plausible faults. The critical set is then given special consideration in developing recommended actions as explained on page 25 under the section entitled Revised Criticality.

Aircraft Configuration

System configuration is an important consideration for any diagnostic aid because as configurations change, the set of valid plausible faults will also change. If the diagnostic aid does not consider changes in system configuration, it will provide incorrect or misleading results. For example, consider an F-16 which is configured completely with conventional weapons but has a nuclear Remote Interface Unit (RIU) installed in one of the pylons. On performing a system BIT, one of the symptoms that will appear on the Fire Control Navigation Panel display is lost communication with the nuclear RIU. However, this error message is normal for the F-16's all-conventional weapons configuration. Due to configuration irregularities, a symptom is present that is normal for the current configuration. If the symptom is considered a valid problem indication, incorrect diagnostic sequencing will inevitably occur.

To avoid such confusing circumstances, the IMIS-DM is notified of the aircraft's configuration during the initialization sequence so that appropriate symptoms are ignored and appropriate faults eliminated from consideration prior to diagnostics.

Evaluation of Faults

The IMIS-DM provides a multifaceted approach to the evaluation of potential faults, combining them into sets using the multiple fault evaluation, partitioning possible faults from unlikely faults and attacking the suspected combinations that most likely are the cause of an inoperable or malfunctioning system. The idea of attacking more than one possible fault at a time is a new development in aircraft diagnostics and is incorporated in all the decision-making algorithms presented in this section.

Multiple Fault Evaluation

"Multiple faults" are two or more faults that occur simultaneously. Multiple faults can appear in a system in a variety of ways. The algorithms employed in the IMIS-DM are designed to handle all types of multiple faults that might occur, including:

1. Single Symptom. This type of multiple fault arises when a single symptom is identified; however, two or more failures might actually be causing that symptom.

2. Multiple Symptom. This is a more complex type of multiple fault. In this case, multiple symptoms are present which can be caused either by a single fault or by a combination of faults.

The IMIS-DM attacks the problem of multiple potential faults by considering several factors: distribution of fault probabilities for the symptoms being considered, how the symptoms span the set of possible faults, the lower probability of independent events occurring simultaneously, and the influence of the time required to complete each possible action. Consider Table 3, a multiple fault scenario including three symptoms and six faults:

Table 3. Fault/Symptom Probability Matrix

	F1	F2	F3	F4	F5	F6
S1	.25	.25	.50	0.0	0.0	0.0
S2	0.0	.25	0.0	.25	.25	.25
S3	0.0	.25	.25	.50	0.0	0.0

The fault weights per each symptom reflect the probability that each fault caused that symptom when the symptom occurs by itself. For example, the probability that F_4 caused S_2 is 0.25. Given all three symptoms occur at the same time, the weights need to be computed to reflect the current situation. Any faults in the intersection now have a higher probability. This higher probability needs to be reflected in the symptoms. For example, F2 has a higher probability of having been the cause of S1 given that S2 and S3 also have occurred. So F1 and F3 have lower probability of having been the cause of S1 and so forth. The revised probabilities can be determined using conditional probabilities:

$$p(A/B) \cup p(A/C) = p(A/B) + p(A/C) - (p(A/B) * p(A/C)) \qquad (10)$$

yields

F1	F2	F3	F4	F5	F6
.25	.578	.625	.625	.25	.25

These are the probabilities the faults caused the current situation. To reflect the adjustment, we now replace the current symptom to fault weights by normalizing the new weights for each symptom, yielding:

	F1	F2	F3	F4	F5	F6
S1	.17	.40	.43	0.0	0.0	0.0
S2	0.0	.34	0.0	.37	.15	.15
S3	0.0	.32	.34	.34	0.0	0.0

As the probability of the intersection goes up, the non-intersection faults probabilities go down. If there is no intersection, the probabilities remains the same. The resulting fault probabilities are used throughout the remaining calculations.

Fault Partitioning through Fault Manipulation

In performing fault manipulation, faults are moved from set to set, presenting new fault combinations to attack, saving fault combinations removed from current consideration, and exculpating potential faults due to passed tests and functional checks. Figure 5 provides a graphic view of fault movement during the diagnostic process.

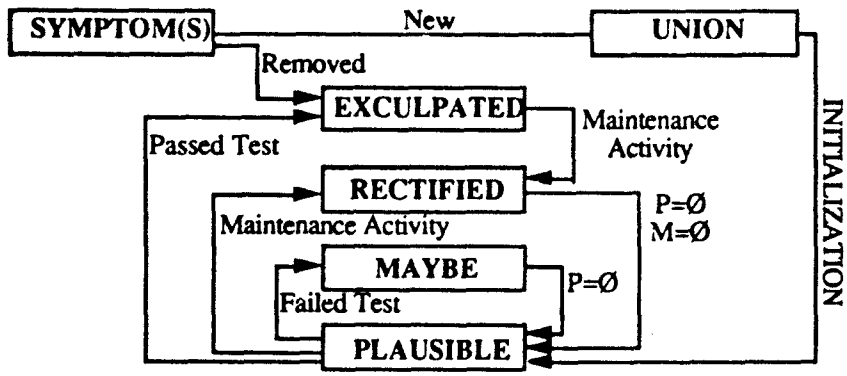


Figure 5. Fault Manipulation.

Upon initialization, the diagnostic module performs a multiple fault evaluation and produces a plausible set of potential fault combinations. The plausible set of potential fault combinations is evaluated and presents a ranked list of options. Based on the option selected and its results, several fault manipulations can occur:

1. Completing a rectification results in spanned faults being placed into the rectified set. This manipulation groups all rectified faults and saves them. If the plausible set is exhausted and the system malfunction still exists, then the rectified set will be used for continued diagnostics. One cannot assume all replacement units from supply are good. Completion of the rectification also prompts a functional check for system health information. If a "pass" on the functional check is observed, all potential faults are exculpated and diagnostics end. Conversely, if a "fail" is observed, the functional check returns updated symptom information. If there are no changes in

the symptom status, diagnostics continue with the current plausible set. If there are changes in the symptom status, the multiple fault evaluation is performed with the new symptom status and a new plausible set is produced.

2. Fault manipulation after a test depends on the test outcome. If a "pass" is observed, potential faults associated with that test are exonerated. They are placed into the exculpated set and eliminated from consideration. Faults in combination with the exculpated faults are placed in the maybe set, provided they are not part of another fault combination in the plausible set. If a "fail" is observed, one or all of the faults spanned by that test are known bad. These faults and their combinations remain in the plausible set for the next action ranking, and all other fault combinations are placed into the maybe set for future evaluation.

When the plausible set has been exhausted, faults are transferred from the maybe set into the plausible. This fault manipulation process repeats until diagnostics are successfully completed or until symptoms are returned and no faults are left in the maybe or plausible sets. When this situation occurs, the faults from the rectified set are placed in the plausible set and diagnostics continue.

Action Ranking

Upon performing the multiple fault evaluation, the diagnostic module uses plausible fault combinations to evaluate the diagnostic actions available for isolating and repairing the aircraft system. A split-half strategy is incorporated into the best test and MOT algorithms to obtain the most information gained per unit of invested time. The best test is then ranked against available actions using the action ranking routines. Hence, the module determines the highest likelihood of fixing or isolating the problem in the least amount of time and cost. Included in action ranking are the dominant action, rectification, and second-step look-ahead analysis.

Split-Half Strategy

The diagnostic module uses a split-half troubleshooting strategy. The initial symptoms' spanned set of potential faults determines the initial plausible set (that set in which at least one fault must exist). Each test's intersection with the plausible set is evaluated. The test that most nearly divides the initial set in half is selected as the best next test. This process is repeated until the plausible set has only one component or until no tests are available that reduce the size of the plausible set. In the latter case, a brute force method of exchanging components is adopted.

A split-half strategy will always isolate a fault in the fewest number of steps whenever test times and fault probabilities are equal. However, test times and fault probabilities are rarely equal; moreover, other constraints also have significant bearing upon the selection of an appropriate diagnostic strategy. Consequently, in developing the action ranking routines, the following additional options were implemented. They are described below.

Best Test	Revised Second-Step Look-Ahead	Estimated Time in Commission (ETIC)
MOT	Revised Criticality	Can Not Duplicate (CND)
Dominant Action	Interleaving Actions	Cannibalization/Facilitate Other Maintenance
Best Rectification	Degraded Mode	

Best Test

The information gained from a binary test is reflected in both the pass result and the fail result. A best test is one which maximizes the information gained from whichever result occurs. Different tests frequently consume different amounts or kinds of resources. That is, there is a cost associated with the choice of a best test. A commonly available metric about which cost can be

allocated is the time to perform the test. Therefore, test time or task time has been used throughout this project as the basic cost metric. In this case, we have chosen to evaluate the best test for maximum information gained per unit of time. Consequently, the best test evaluation used in this program has been defined as the following:

$$BT = \max \frac{I_j}{T_j} \text{ , where} \tag{11}$$

$$I_j = \frac{\sum_{i=1}^{PS} FR(1)}{FR(PS)} * \frac{\sum_{i=1}^{PS} FR(0)}{FR(PS)} \tag{12}$$

Tj = time to accomplish test j.

The tests to be ranked against one another are determined by the usefulness of the tests. We can say a test is useful in the following manner. Given a set of faults, the faults are compared against two rules:

1. At least one but not all of the faults in the set is contained in the plausible set.
2. The set of faults may not possess, as a subset, all the implicated faults of any of the observed symptoms.

If the set of faults conforms to both of these rules, then it will be deemed useful. For binary tests, if the spanned faults of the test are useful, then the test is considered useful. For MOTs, only one of its outcomes' spanned faults needs to be useful to consider the test useful. After all the applicable tests are checked for their usefulness, the tests deemed useful will be ranked against one another. For example, given a system model as shown in Table 4, the test usefulness is as shown in Table 5.

Table 4. System Model for Test Usefulness

	S1	S2	S3	T1	T2	T3	T4	T5	T6	T7	T8
F1	1	1	1	0	0	0	1	1	1	1	1
F2	1	1	0	0	0	1	0	0	0	1	1
F3	1	0	0	0	0	0	0	0	0	0	1
F4	0	0	1	1	0	1	0	1	0	1	1
F5	0	0	0	0	1	0	0	0	0	1	0

Plausible Faults (F1, F2, F3, F4)

Table 5. Test Usefulness Demonstration

Test	Useful	Why Not
T1	Yes	
T2	No	Fails rule 1
T3	Yes	
T4	Yes	
T5	No	Fails rule 2 for S3
T6	Yes	
T7	No	Fails rule 2 for S2 and S3
T8	No	Fails rule 1

Multiple Outcome Test (MOT)

A test which has MOTs creates special problems when trying to measure its worth against other available tests that also split a plausible set. The problem is because a test with multiple outcomes is not binary (a pass is not the complement of a fail). A purely binary test will result in the operation of all spanned faults in the event of a pass and the inclusion of all spanned faults in the event of a fail. Conversely, a test with multiple outcomes, an MOT, would operate all spanned faults in a pass condition but include only a restricted number of spanned faults in one of several possible fail conditions. Any one of several possible fail results can lead to isolate faults to a number much smaller than the test's spanned set. Therefore, an MOT is generally more powerful than a binary test which spans or splits the same set; hence, it is more valuable.

A perfect MOT is one in which each outcome isolates a single fault, and there are sufficient outcomes so that each fault spanned by the MOT can be isolated, as shown below:

	SPAN
	0 0 0 0 0
	1 0 0 0 0
OUTCOMES	0 1 0 0 0
	0 0 1 0 0
	0 0 0 1 0
	0 0 0 0 1

Conversely, a poorly designed MOT would neither isolate single faults nor contain sufficient outcomes, as follows:

	SPAN
	0 0 0 0 0
OUTCOMES	0 1 1 1 1
	1 1 1 1 0

From the above, it can readily be seen that the value of an MOT is related to the "sparseness" of dependencies coupled with the number of possible outcomes. Consequently, it was our intent to develop a relationship that takes advantage of these logically and aesthetically

obvious relationships. The best test from a set of binary tests can be determined from equation (11); hence, the logical approach to accommodating MOTs was to operate directly upon the best test algorithm (repeated in equation (13)).

$$BT = \max \frac{1}{T} \times \frac{\sum_{i=1}^{PS} FR(1)}{FR(PS)} X \frac{\sum_{i=1}^{PS} FR(0)}{FR(PS)} \tag{13}$$

Examination of the best/worst MOT displays showed that a scale factor that describes the "sparseness" of dependencies can be readily determined from the ratio where the zeros and ones are counted for all test outcomes.

$$R = \frac{\sum_{rows} 0's}{\sum_{rows} 1's} \tag{14}$$

In addition to looking at the relative efficiency of the test through the R ratio, we must also evaluate the value of the individual test outcomes. This task can be accomplished using the same algorithm for best test and then computing the average for all test outcomes as shown below:

$$\bar{I} = \frac{1}{n} \sum_{j=1}^{n} \left[\frac{\sum_{i=1}^{PS} FR(1)}{FR(PS)} \times \frac{\sum_{i=1}^{PS} FR(0)}{FR(PS)} \right] \tag{15}$$

where $\bar{I}$ = the average information gain from the MOT, and
n = number of outcomes.

The revised best test algorithm then becomes the following:

$$BT = \max \frac{R\bar{I}_j}{T_j} \tag{16}$$

Having established this algorithm as an accurate measure of the value of a MOT, it was then necessary to look at how this algorithm affected the result from evaluating a binary test. If R and n are defined as one for binary tests, then the above equation reduces to the original best test algorithm. Hence, this solution to the MOT problem allows a single equation to be used to compute all best test values.

Dominant Action

Given a particular symptom or set of symptoms, the plausible set may contain a particular component that is so predominantly likely to be the cause that an immediate rectification action is warranted. This sort of alternative may be particularly attractive under certain criticality considerations. In these cases, resource conservation can become a secondary consideration.

To examine this change of philosophy, we needed to establish a mathematical relationship that measured rectification time against the choice of methods.

Let: RT = Rectification Time
TT = Test Time
RRT = Removal and Replacement Time
PFn = Probability that fault n has occurred

Assume there are two strategies available. Strategy 1 is to perform an immediate replacement of the most likely cause of the faults in the plausible set without any attempt at fault isolation; then, if necessary, perform a fault isolation test if one is available. Strategy 2 is to perform diagnostic testing to isolate the faulty component first. The choice between Strategy 1 and 2 can be made on a particular component based on the following dominant action equation.

$$\begin{aligned} \text{Delta RT} &= \text{RRT} - (\text{RRT} \times \text{PFn} + \text{TT} \times \text{PFn}) \\ &= \text{RRT} - (\text{RRT} + \text{TT}) \times \text{PFn} \end{aligned} \tag{17}$$

This formula determines the difference between the time to rectify the component (RRT) and the time to perform the test (TT) first plus rectify (RRT) that component using the probability (PFn) that fault n occurred.

Therefore, if Delta RT < 0, Strategy 1 (swap first) is the best option, signifying that the probability of that component being faulty is so high that it should be replaced without testing and/or testing time is high compared to rectification time. If Delta RT > or = 0, Strategy 2 (test first) is the best option because the probability of that fault having caused the problem is not very likely and/or the rectification time is very high, and it is better to test before rectifying.

Under pressing time considerations, the dominant action recommendations will generate a fixed component in minimum time. However, if test times approach or become large compared to replacement times, the equation yields a swap-first decision with a decreasing probability that the swap action will fix the fault. Such a situation could be very inefficient when there are no pressing time considerations or there are few spare components. To solve this problem, the Second Step probability of success was developed. It provides an examination of what the maintenance technician could expect to face at the end of the second upcoming maintenance event in the diagnostic sequence. This analysis is discussed under Second-Step Look-Ahead.

Best Rectification

If no tests are available, diagnostics must be completed by rectifications alone. The best rectification routine recommends the best rectification to perform first based on time to rectify and probability of failure. This routine is also performed to provide the maintenance technician with the human interface feature of a displayed ranked list of the five best actions.

The best rectification (BR) analysis provides a strategy to minimize the total time to system rectification. The following variable definitions apply:

RRT_i = Remove and Replace time of the ith component considered by the plausible set of faults.
RRT_T = Sum of Removal and Replacement times for all components considered by the plausible set of faults.
FR_i = Failure rate of a given fault or component = 1/MTBF
FR_T = Failure Rate of the plausible set.

The results of the analysis are generalized to a multicomponent system if we recognize that individual comparisons provide only a relative ranking between components. Failure to achieve

success on the first option requires recycling through the algorithm to determine the next best option. Therefore, the generalized form of the algorithm to rank the trade-off between failure rate and substitution time is as follows:

$$BR_i = \frac{RRT_i}{RRT_T} - \frac{FR_i}{FR_T} \qquad (18)$$

for the ith component.

Results from this analysis will range from -1 to 1. The component with the lowest BR_i is the optimum candidate for the best rectification. All the actions are ranked from lowest to highest for ranked rectification list.

Second-Step Look-Ahead

The second step look-ahead analysis provides a diagnostic recommendation based on the cost difference between the dominant action and the best test by analyzing what the maintenance technician could expect to face at the end of the second upcoming maintenance event (next activity) in the diagnostic sequence. Upon completion of the dominant action analysis, the dominant action recommendation can take two routes depending on the state of system criticality. If criticality is invoked, the diagnostic module automatically recommends performing the dominant action because cost is not a factor if the system is to be repaired in minimum time. If the system is not deemed critical for the next mission, the second step look-ahead analysis is performed.

When second-step look-ahead is chosen, two viable diagnostic activities, a best test and a dominant action, are available to continue diagnostics. The unit cost of the dominant action and the best test are calculated using the formulas given below. The activity with the lowest cost is recommended as the next activity to be performed.

To correctly perform this analysis, one must realize that a test cannot fix a system (only isolate faults). Therefore, the probability of fixing a system by performing a test is zero. Likewise, tests do not require any units from supply (UFS), so UFS, as a result of test performance, is 0. To perform each analysis, the dominant action and best test cost analyses, the diagnostic module calculates UFS, time, and probability of success (POS) associated with the performance of the current activity under investigation and the next best activity. These calculations are then used to formulate the cost of each activity. The activity that exhibits the least cost is recommended.

Let:

$PSSS$	=	The probability of success of the next best activity (second step).
PS_{DA}	=	The probability of success of the dominant action.
RRT_{DA}	=	The time required to perform removal and replacement of the dominant action.
BTT	=	The time required to perform the best test.
NAT	=	The time required to perform the next best activity (second step).
Time	=	The time to complete an activity normalized by its probability of success.
UFS	=	The units from supply used to perform the activity.
POS	=	The probability of success by performing the current activity and the next best activity (second step).
PTO_i	=	The probability of ith test outcome i.
$PSSSTO_i$	=	The probability of success of the next best activity (Second Step) based on test outcome i.

n = The number of test outcomes, which for a binary test would be 2 and for an MOT many.

a. Dominant Action Cost:

$$UFS = 1 + (1 - PS_{DA}) \quad (19)$$

Where $(1\text{-}PS_{DA})$ is equal to 0 if the next best activity is a test (no UFS for a test).

$$Time = RRT_{DA} + [(1 - PS_{DA}) \times (RRT_{DA} + NAT_{SS})] \quad (20)$$

$$POS = PS_{DA} + PS_{SS} \quad (21)$$

Where PS_{SS} is equal to 0 if the next best activity is a test (tests cannot fix).

$$\text{Rectification Cost} = \frac{(UFS \times Time)}{POS} \quad (22)$$

b. Best Test Cost:

$$UFS = \sum_{i=1}^{n} (1\,unit \times PS_{SSTO_i}) \quad (23)$$

Where PS_{SSTO_i} is equal to 0 if the next best activity is a test (no UFS for a test).

$$Time = [\sum_{i=1}^{n} (NAT_{SS} \times PTO_i)] + BIT \quad (24)$$

$$POS = \sum_{i=1}^{n} (PS_{SSTO_i} \times PTO_i) \quad (25)$$

Where PS_{SSTO_i} is equal to 0 if the next best activity is a test (test cannot fix).

$$\text{Test Cost} = \frac{(UFS \times Time)}{POS} \quad (26)$$

This concept, when expanded to the general case, proved to be far more valuable than choosing a strategy based solely on time. Furthermore, the idea of cost could be readily expanded if the necessary data to clearly allocate costs associated with procurement, storage, transportation, maintenance, and test equipment for competing LRUs in the algorithm were obtained. Lacking such sophisticated data, a simple supply parts count can be an effective measure of the cost of a test versus an action decision.

Revised Criticality

The criticality function has been designed for situations when operational demands prevent maintenance practices that take a minimum time to repair a malfunctioning system. Under some circumstances, it is possible operational requirements would declare a system usable if it can be determined the fault present in the system is in a part of the system not essential for the next sortie.

This circumstance is recognized in the Air Force by the current practice of declaring a system Partially Mission Capable (PMC). When this practice is invoked, the criticality function of the IMIS-DM can be set to accommodate a change in maintenance practices.

The diagnostic data allow faults to be assigned to certain systems and subsystems that may or may not be designated as critical. If a system or subsystem has been declared critical, then the diagnostic module is modified to make recommendations based upon a criticality algorithm. The criticality algorithm is designed to allow a critical fault decision at the earliest possible time in the maintenance process. In developing the algorithm, the following definitions were posited.

1. Critical Test - A test that examines all potential faults declared critical.

2. Critical Rectification - A rectification that acts to repair all faults declared critical.

The following assumptions apply in the development of the criticality algorithm.

1. Operational considerations can result in the imposition of critical requirements.

2. The faults comprising a critical set for a given critical function can be identified in the data base.

3. The value of an operational hour to the operations community is at least as great as the value of a maintenance labor hour to the maintenance community.

4. The tests available in the critical set of faults are at least as effective as the tests available in the full system set of faults.

Equations were developed to account for criticality in the probability of the fault sets and multiple outcome tests. Each test's outcomes can create a fault set. For the pass outcome of each test, the fault set will be the unspanned plausible faults of the test. For a fail outcome, a fault set will consist of the plausible spanned faults of the outcome. To obtain the probability for the pass outcome fault set, the probability of the plausible spanned faults is used. For fail outcome fault sets, the probability of the outcome will be used.

The steps to isolate faults in each fault set are conditioned using the probabilities formulas. The end result is the sum of the steps for the pass outcome and all the fail outcomes. The final forms of the equations in the criticality algorithm of the IMIS-DM were:

$$\text{STREP} = \left[(1\text{-Pfb}) * \text{Log}_2(2*\#\text{nsb})\right] + \left[\sum_{i=1}^{\#\text{fob}} \text{Pfob}_i * \text{Log}_2(2*\#\text{sfob}_i)\right] + 1 \qquad (27)$$

$$\text{STFLY} = \left\{\sum_{i=1}^{\#\text{foc}} \left[\text{Pfoc} * \text{Log}_2(2*\#\text{sfoc}_i)\right]\right\} + 1 \qquad (28)$$

$$\text{STTOT} = \left[(1\text{-Pfc}) * \text{Log}_2(2*\#\text{nsc})\right] + \left[\sum_{i=1}^{\#\text{fob}} \text{Pfoc}_i * \text{Log}_2(2*\#\text{sfoc}_i)\right] + 1 \qquad (29)$$

Where,

Pfb	= the probability the best test will fail,
#nsb	= the number of plausible faults nonspanned by the best test,
$Pfob_i$	= the probability of fail outcome i of the best test,
#fob	= the number of fail outcomes for the best test,
fob_i	= fail outcome i of the best test,
$\#sfob_i$	= the number of plausible spanned faults from outcome i of the best test,
Pfc	= the probability the critical test will fail,
#nsc	= the number of plausible faults nonspanned by the critical test,
$Pfob_i$	= the probability of fail outcome i of the critical test,
#foc	= the number of fail outcomes for the critical test,
foc_i	= fail outcome i of the critical test, and
$\#sfoc_i$	= the number of plausible spanned faults from outcome i of the critical test.
STFLY	= the steps to declare a system ready for operations when a critical test is performed as the first test, and
STREP	= the steps required to repair a system when the most appropriate test is performed first,
STTOT	= the steps to perform maintenance required to bring the system to operational condition after a critical test.

Interleaving Actions

The IMIS-DM algorithm, which evaluates tests and actions, is also used to generate a ranked list of options. The best test and dominant action loops already performed the basic evaluating and ranking functions, and, with minor modification, were broadened to include the interleaving actions facility.

The first step after entering the loop is to initiate the multiple fault algorithms to generate a ranked list of fault sets, which represent the rectifications against which tests will be ranked. Next, using the methodology outlined for selecting a best test, the diagnostic module performs analyses and selects best tests for the given information and ranks these tests in decreasing order. The dominant action equation computes times to accomplish the rectification versus the best test and provides a decision whether to test or replace with the dominant action. The second step look-ahead analysis is then performed if any of the ranked actions are dominant and criticality is not invoked. If criticality is invoked, the dominant action is chosen. The test or action chosen becomes the first option in the list of interleaved actions. It is then removed from further comparison. The tests or actions not chosen are evaluated for the next interleaved option. This process of comparison using the best test, dominant action, and second step look-ahead analyses continues until the list of interleaved tests and actions has five entries, at which time the routine is terminated. For example, Action B is selected the first time as dominant action, and is chosen over Test 1, then Action B goes to the top of the list. It is removed from consideration and the comparison is executed again. If Test 1 dominates over all actions the second time, then it is placed on the list below Action B. It is then removed from consideration, and the loop is executed again. The third time, Test 2 is compared to all actions. Whichever activity dominates will be placed below the last activity placed on the list. A possible display of the top five options would be in the following format:

1. ACTION B
2. TEST 1

3. ACTION C
4. ACTION A
5. TEST 2

Degraded Mode

Degraded mode occurs when the diagnostic module can no longer recommend an action and all suspected faults, given symptom occurrence, have been eliminated from consideration. This situation can happen if the diagnostic module is given incorrect or incomplete data. At this point, the diagnostic module recommends transition to the degraded mode.

The objective of the degraded mode is to find a test that fails resulting in a plausible set of faults, and/or do a rectification that passes a system health check. In certain situations, the technician may choose to put the system into degraded mode in one of two ways. The technician can select degraded mode if he decides the diagnostic module is no longer helping in troubleshooting. Or he can select degraded mode in response to a recommendation from the diagnostic module.

The diagnostic module will recommend degraded mode of operation if:

1. a symptom is present but all suspected faults that could have caused the symptom are eliminated from consideration by either passed tests or by the correction of another symptom; or,

2. rectifications have been performed for the second time on the same components where the fault is suspected; or,

3. a symptom is confirmed present and is not in the data base.

Although the maintenance technician has full control of degraded mode selection, degraded mode is necessary to continue diagnostics when:

1. the diagnostic module recommends the degraded mode based on the above occurrences;

2. CND/Intermittent (unverifiable starting points) messages are received; and,

3. whenever the technician chooses, i.e., the technician does not want to follow any of the diagnostic module's recommendations.

When degraded mode is selected by the technician, the physical and functional assessment modules are suspended by the diagnostic controller module and a message appears notifying the technician he has entered degraded mode. This message remains on the screen at all times during degraded mode assessment. To aid the technician, a smart Table Of Contents (TOC) is created. The smart TOC consists of two lists. The first list contains ranked rectifications based on component MTBFs. The components with the lowest MTBFs are ranked highest on the list. The second list contains an ordered list of tests based on probability of failure, calculated by summing the failure rates of all spanned faults. Moreover, to provide a more accurate test ranking, any information gained during physical and functional assessment is used to alter the test's failure probabilities. When modifying the test's failure probabilities, exculpated faults are not used in computations of tests containing them.

When the smart TOC is displayed, the technician can select a test or rectification from the ranked lists and perform the selected action. Depending on the action performed and its results, diagnostics can proceed in several ways. At the completion of any action, the maintenance

technician must either suspend diagnostics, select another action from the TOC, or exit the degraded mode. The technician must consider the precedences described below, reflecting logical continuation of isolation and repair.

1. If a test is selected during degraded mode assessment and a fail result is exhibited, a new plausible set is established. The degraded module then records the new plausible set of faults for further isolation and repair. Once a new plausible set of faults is established, degraded mode can be exited and physical or functional assessment can proceed from the new plausible set.

2. If a test is selected during degraded mode assessment, and a pass result is exhibited, the degraded module records the exculpated faults, eliminates the performed test from consideration, and reranks the TOC's test list. The maintenance technician can then select and perform another action from the TOC.

3. If a rectification is selected and performed during degraded mode assessment, and changes to the system are exhibited as a result of a functional check, the degraded module records the appearance of new symptoms and/or existing symptoms are modified or deselected. Given the new set of observed symptoms, degraded mode can be exited and diagnostics can proceed with physical or functional assessment.

4. If a rectification is selected and performed during degraded mode assessment, and no changes to the system are exhibited as a result of the functional check, the degraded module records that rectification, eliminates the performed rectification from consideration, and reranks the TOC's rectification list. The maintenance technician can then select another action from the TOC.

Hence, the diagnostic module can continue fault isolation and repair when faced with incorrect or missing data.

Estimated Time In Commission (ETIC)

The ETIC function provides a calculation of the estimated time a diagnostic exercise will be completed and the weapon system declared operational. ETIC can be calculated several ways. In the manual world of TOs prevalent today, the maintenance technician makes an estimate based on experience of the time it will take until he completes the first rectification activity he feels he will have to perform to resolve the reported symptom. This estimate is provided with little more knowledge of the probability of success than, "This sort of worked last time."

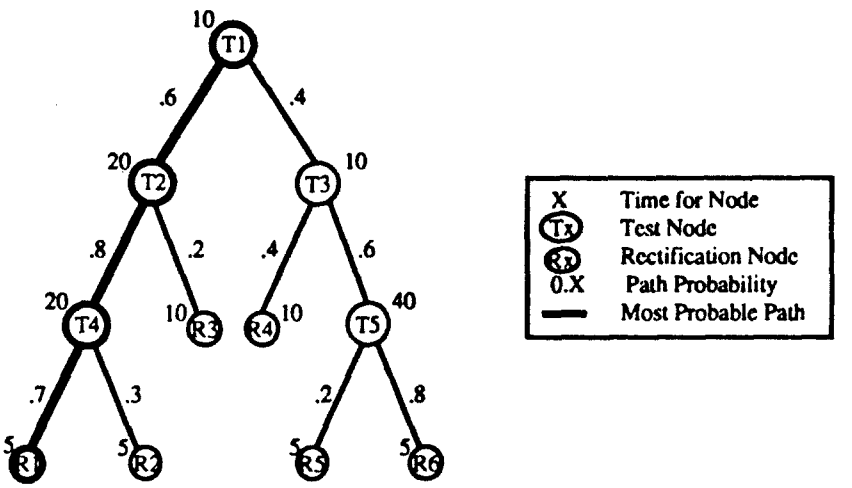
In the IMIS-DM, some rudimentary probability calculations are available; hence, we might improve on the current estimating procedure. Two methods of performing the calculation are readily apparent. We might choose to perform an exhaustive look-ahead function in which all possible paths to fault isolation and repair are completed. The time to achieve completion of each root node in the tree could be accumulated and the probability of having to proceed to that particular node could be calculated. Given these calculations, ETIC could be presented as a probability table showing the estimated time to achieve some predetermined probability of success. This is shown in Table 6.

Table 6. Estimated Time in Commission (ETIC)

Probability	0.25	0.50	0.75	0.95
ETIC	1.0 Hrs.	1.5 Hrs.	2.2 Hrs.	4.8 Hrs.

The second method is to modify the exhaustive look-ahead calculation to account solely for the most probable path to fault isolation and repair. In this methodology, only one path is followed to the root node. That is the path most likely to be chosen at each succeeding node in the tree. The result would be a time and probability to the most likely rectification given the current symptom. This procedure is illustrated in Figure 6.

The method finally selected was a combination of these two methods. The time to completion of the most probable path is calculated along with the probability of following that path, as in the second method. However, rather than stop at that point, other paths through the tree are traced until the cumulative time to proceed further down that alternative path exceeds the time to most probable path completion. For any root node reached prior to exceeding the most probable path time, the probability of reaching that root node is added to the probability associated with the time in commission. In this manner, we achieve a much more accurate estimate of the probability that the weapon system will be repaired at or before the ETIC established for the most probable path. This process is illustrated in Figure 7.



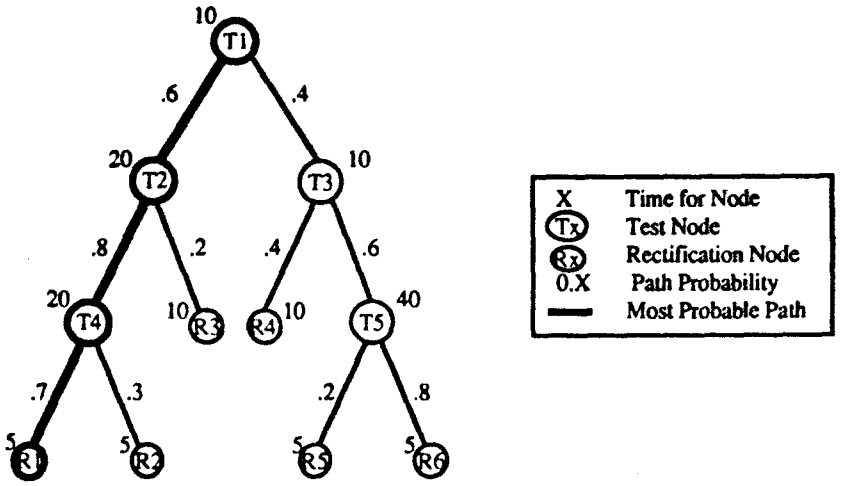
Most Probable Path (T1, T2, T4, R1)
ETIC: 55 Prob: 0.336

Figure 6. Most Probable Path Calculation.

Can Not Duplicate (CND)

A CND problem exists when an operator reports a problem but a fault verification test on the system fails to reveal any problem. This situation can result for any number of reasons, but two diametrically opposed reasons create a diagnostics problem. The problem reported may in fact have been transient and the problem no longer exists in the system. In this case, there is no fault and any maintenance activity to "fix" the problem is probably wasted effort and may in fact be more likely to introduce errors than to fix them. The opposing problem is that the fault verification test did not span the fault present in the system or the test did not reproduce the environment in which the problem occurred. In this case, a fault does exist in the system, but the technician has no way to observe the effect and begin fault isolation procedures. There is no convenient method to determine which of these two possibilities exists after a fault verification test passes in all respects.

The revised IMIS-DM handles this problem by using an aircraft history file stored in the PMA and a predetermined CND strategy. The strategy employed is based on the concept of repeat and recurring problems. A repeat problem is a one that occurs again on the next flight after it was signed off by maintenance as CND following the previous flight. A recurring problem is one that reoccurs during the second or third flight following the initial report. An occurrence on a fourth or subsequent flight is treated as an independent, unrelated occurrence of the problem.



Most Probable Path (T1, T2, T4, R1)
ETIC: 55 Prob: 0.76

	Time	Probability
R_1	55	0.336
R_2	55	0.144
R_3	30	0.12
R_4	30	0.16
R_5	>55	N/A
R_6	>55	N/A

Figure 7. Modified Most Probable Path Calculation.

The aircraft history file maintained on the portable display device contains recent aircraft flying history (the last four flights), to include (a) date flown, (b) system symptoms, and (c) corrective action taken for any reported symptom. Using this aircraft history file, the IMIS-DM CND handling strategy is as follows:

1. For the initial occurrence of a CND in a system, treat the problem as transient and enter the corrective action as CND.

2. For a repeat occurrence of the same or a closely related symptom:

   a. search for and perform any tests not included in the fault verification sequence, but span faults implicated by the reported symptom.

   b. if no tests are available to do the above, perform the action at the top of the Best Actions list.

c. if tests meet the above criteria, but all pass, then perform the action at the top of the Best Actions list.

d. if tests fail, then perform normal diagnostics for fault isolation and repair.

3. For a third and subsequent sequence similar to 2 above, perform the second and subsequent recommended actions on the Best Actions list.

Cannibalization/Facilitate Other Maintenance

Cannibalization Modeling. The IMIS-DM can differentiate between feasible options and those that are not due to availability criteria. If the user selects an option that cannot be accomplished because of unavailable parts, the option is still displayed, but in reverse video, denoting the parts not available. If the technician selects this option, he receives a warning that the needed parts are unavailable but is allowed to proceed without further assistance. This enhancement assists the technician when parts are not available to perform the recommended action. This process can be thought of as cannibalization modeling, because the technician who selects actions for which parts are unavailable is probably planning on taking parts from a "good" aircraft to complete the maintenance action. By modeling this process within IMIS-DM, several facets of the maintenance process are expedited: (a) crucial maintenance can be performed to bring a system to operational status even in the case of supply shortages, (e.g., combat); (b) functional testing is facilitated to promote fault isolation; and (c) considerable time is saved.

Cannibalization Process. Cannibalization occurs when the technician removes "good" parts from an aircraft for use in another aircraft under repair. This action can be likened to a swap action, the major difference being that the swapped part is coming from another plane rather than supply. During this swap action, TOs are needed for removal and replacement of parts for the plane being cannibalized as well as the plane under repair. The IMIS-DM will assist the technician in this process by guiding him through the cannibalization, providing appropriate TOs in a logical order and also allowing him to choose how the cannibalized part is to be used. Two cases are possible. The first is one where the swapped part is to remain in the aircraft under repair. This case occurs when there is a pressing need to bring the plane under repair to operational status, and disabling another plane is an acceptable consequence. The second case occurs when the swapped part is needed only for troubleshooting purposes. In this case, the swapped part will ultimately be returned to the plane from which it was taken. Each case requires different action by the IMIS-DM.

Cannibalization Dialogue. To incorporate this facility into the IMIS-DM, a dialogue "tree" of possible paths the technician might take during cannibalization was developed and used as the baseline for implementation. This dialogue includes the steps associated with each possible cannibalization case and queries the user as to his intent, to display the appropriate TOs. The cannibalization routine is initiated when the user selects an option displayed as requiring parts which are unavailable. When this occurs, the IMIS-DM displays a warning stating the selected option is unavailable and asks the user if he wishes to cannibalize another aircraft for the necessary part. After viewing the warning screen, the user may choose to initiate the cannibalization sequence with a YES response, progressing through the shown sequence, or he may avoid cannibalization with a NO response which will proceed with the repair as normal as though the part were available. If the user chooses to cannibalize another aircraft, three sets of TOs are displayed in sequence for removal of the "good" part from the cannibalized aircraft, removal of the "bad" part from the plane under repair, and replacement of the good part in the aircraft under repair. At this time, a functional test is performed to evaluate the impact of the good part. If the functional check fails, further removal and replacement of these parts are suspended until the system checks out. Once the system check is OK, TOs are displayed to return parts to their original location. If the functional test passes, the IMIS-DM prompts the user for his intentions: troubleshooting or to ready an aircraft.

Depending on the response, the IMIS-DM displays appropriate TOs to facilitate that choice. When any of these three paths is completed, diagnostics are resumed. This process is shown in Figure 8.

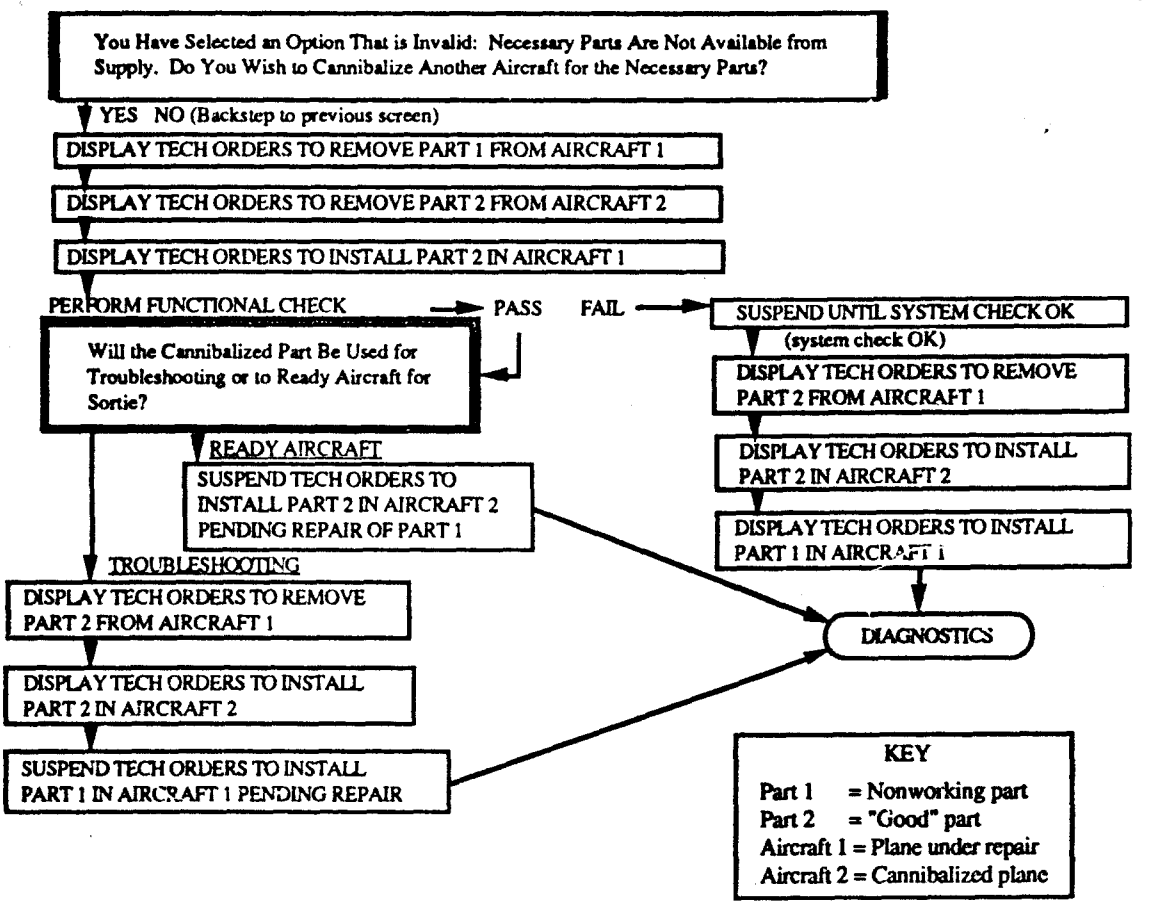


Figure 8. IMIS-DM Cannibalization Dialogue Flow.

Reinitialization/Change in Symptom

The IMIS-DM can react to changes in the diagnostic situation by updating parameters during diagnostics. Changes in symptoms might occur if a symptom is discovered or removed during rectification. As the diagnostic module is executed and as the technician applies the information to the problem, certain information is gained. Tests are passed/failed, and faults are exculpated from the plausible set. This information is useful to the diagnostic module because it reduces the problem's complexity and brings the solution closer.

For example, assume the IMIS-DM begins diagnostics with a set of symptoms implicating a given number of faults. Symptoms are eliminated as faults are isolated and components rectified. Assume a specific symptom has been eliminated and, with it, several faults are removed from consideration. There still remain other symptoms and faults to be removed; however, the problem's complexity may be reduced. One of the exculpated faults might be implicated by one of the remaining symptoms. By knowing that this fault is exculpated, the plausible set of faults for the

symptom still being investigated is reduced and the resulting computations are simpler and quicker. The ability to account for a change in symptoms is important if the diagnostic module is to effectively attack a problem.

Whenever a rectification or maintenance action is completed, the module performs a system check and notifies the technician of any remaining symptoms. Any changes in the state of the fault/symptom matrix are updated by user inputs and the diagnostic module simply adds or deletes information as necessary. Information is not lost, and any changes in the state of the problem are handled and incorporated in the succeeding diagnostic steps. Data are input throughout the process; the loop is never exited. User input menus to identify symptom changes within the diagnostic loop provide the diagnostic module with a recursive network that is reinitialized at the start of each diagnostic sequence iteration.

V. DIAGNOSTIC ENHANCEMENTS

Enhanced Diagnostic Module Functions

The enhancements described in this section have created a more efficient and accurate diagnostic module. The module now performs degraded mode and revised critical fault assessment, and captures information gained from previously failed tests. It also considers dependent symptom occurrence and time saved for accessing groups of components or LRUs for testing and repair. Other enhancements to the diagnostic module allow changes to test and functional check outcomes in case of incorrect entry of results.

Failed Faults from Previous Test

Under unusual circumstances, earlier versions of the diagnostic module could lose fault isolation information due to fault combination manipulations. A new type of fault list is used in the enhanced version to correct this problem. This new list type is called the isolated faults list. Whenever the plausible fault list contains only one fault, the fault is placed in the isolated faults list before other processing is done. When the plausible set is rebuilt, the isolated faults list is searched for the first isolated plausible fault. If one is found, all other faults are moved to the maybe set. Because the fault is the only one in the plausible set, the module will recommend its rectification with a 100-percent probability.

Access Group

An access group is a group of components unveiled by removing a panel or cover. When ranking tests or rectifications, a diagnostic advisor should consider access group factors for rectification and testing time efficiency. Diagnostic efficiency may be gained when actions performed on access groups reveal more fault-associated components but have high access times. Previous IMIS-DM versions did not consider access times in the ranking of tests and rectifications. Access times in previous versions were assigned to each individual action and were not considered for a commonly accessible group of actions.

The method of approach used to develop this capability was essentially the same as that used to develop an MOT evaluation capability described in the report by Cooke et al. (1990a). The access group algorithm is designed so that once access is gained, each test in the group can be accomplished as though no access time is required. In addition, the best test evaluation in these circumstances is merely an extension of the current best test algorithm. This feature was created by adding an enhancement factor to the best test that accounts for the additional fault isolation capability obtained by gaining access.

The enhancement factor used is:

$$EF_j = \frac{\sum_{k=1}^{N}\left[\frac{\sum_{i=1}^{PS} FR(1)}{FR(PS)} \times \frac{\sum_{i=1}^{PS} FR(0)}{FR(PS)}\right]}{N \times \Sigma T} \qquad (30)$$

Where EF_j = the enhancement factor for test j,

$FR(PS)$ = the sum of all the failure rates for faults in the plausible set,

$\sum_{i=1}^{PS} FR(1)$ = the sum of the failure rates for spanned faults for a given test (T) in the plausible set of faults,

$\sum_{i=1}^{PS} FR(0)$ = the sum of the failure rates for unspanned faults for a given test (T) in the plausible set of faults,

$\sum_{k=1}^{N}$ = the sum of the products of spanned and unspanned tests within the access group,

ΣT = the sum of all test times including access time for the group, e.g., for an access time of 10 minutes creating access to three five-minute tests, $\Sigma T = 25$, and

N = the number of tests in the access group.

The enhancement factor is set to zero if no additional tests are included in the access group. Hence, the best test algorithm used in the IMIS-DM after this enhancement is:

$$BT = \max \frac{R_j \bar{I}_j}{T_j} + EF_j \qquad (31)$$

Where,

BT = Best Test value,
R_j = sparseness ratio of the test span,
$\bar{I}j$ = the average information gain, and
T_j = time to accomplish test j.

"But Not" Data Entry

The "But Not" algorithm implemented in the redesigned diagnostic module retrieves information from test results in the form of observed outcomes and spanned faults, and determines what faults are implicated and exculpated based on the test performed and outcome observed. Previous versions of the diagnostic module did not include "But Not" data entry logic when manipulating faults from test results. The exclusion of this "But Not" data entry logic resulted in inefficient fault isolation and repair decisions because suspected faults remained under investigation after they could have been eliminated from consideration.

The "But Not" algorithm operates on symptoms and MOTs. The presence or absence of symptoms in conjunction with other symptoms can result in implication of reduced fault sets. MOTs have one pass outcome and two or more fail outcomes. There are three types of MOT tests: (a) Complete And Enter One (CAEO), (b) Complete And Enter All (CAEA), and (c) Exit At First Failure (EAFF). The CAEO MOT is completed in full and only one outcome can be entered upon completion of the test. However, CAEA MOTs are also completed in full but all observed outcomes are entered. EAFF MOTs are only completed to the point at which the first failure is observed and at that point the observed outcome is entered.

Each outcome exhibited from a test result maps to a set of spanned faults, exculpated and/or implicated. Binary (BIN), CAEO, and CAEA tests have one pass outcome that, when observed, exculpates all faults spanned by the pass outcome. When a fail outcome(s) is observed from these tests, the diagnostic module implicates and exculpates all faults for the observed fail outcome(s) and then exculpates all the implicated faults of the non-observed outcomes. The EAFF also exculpates all faults of the observed pass outcome. But, if a fail outcome is observed the "But Not" algorithm exculpates all implicated faults for prior non-observed outcomes in the performed sequence and implicates and exculpates faults of the observed outcome. Because of the implementation of the "But Not" data entry logic, known good faults are exculpated while suspected faults are isolated and repaired.

Account for TOC Actions

Earlier versions of the diagnostic module did not effectively allow choices to be made from the TOC. This limited the maintenance technician's ability to perform tasks he considered pertinent but were not in the interleaved actions list. Accounting for TOC actions was implemented easily within the diagnostic module. Now, whenever a TOC test or rectification is chosen, the diagnostic module only needs to be informed the action selected was not from the interleaved actions list. If a test is selected, the observed test outcomes need to be passed to the diagnostic module as well. Furthermore, if the task pertains to any of the existing diagnostic groups, that group will be updated. If no appropriate diagnostic group exists, one is created.

Change Test Result

If a maintenance technician erred in the selection of a test outcome, earlier diagnostic module versions would not allow correct entries to be made easily. The previous actions list is used to ease changing a test's result. The previous actions list contains a list of all the tests and rectifications previously performed. It also keeps a record of the machine's diagnostic state before the action. Once a previous test is selected, the new results are passed to the diagnostic module. The diagnostic group pertaining to that test is removed and replaced with the copy stored with the previous action. If no diagnostic group exists, the copy stored with the test is added to the list of diagnostic groups. The new test results are passed to the diagnostic group. The next item in the previous actions list belonging to the same diagnostic group as the test is updated with the new diagnostic group. Then the diagnostic group is updated with the results of the action. This process is repeated until the previous actions list is exhausted.

Change Functional Check Result

Previous versions of the diagnostic module would not allow a maintenance technician to change manually entered functional check results. If a maintenance technician erred when selecting functional check results, the diagnostic module would continue diagnostic evaluation with incorrect system information and proceed to an incorrect isolation and repair decision. The previous actions list is accessed so the technician can change functional check results. The previous actions list contains a list of all the functional checks, tests, and rectifications previously performed by the maintenance technician, and records a copy of the machine's diagnostic state before each action. Once a previous functional check is selected, the new results are passed to the diagnostic module. The diagnostic group pertaining to that functional check is removed and replaced with the copy stored with the previous action. If no diagnostic group exists, the copy stored with the functional check is added to the list of diagnostic groups. The new functional check results are passed to the diagnostic group. The next item in the previous actions list belonging to the same diagnostic group as the functional check is updated with the new diagnostic group. Then the diagnostic group is updated with the results of the action. This process is repeated until the previous actions list is exhausted.

Enhanced Diagnostic Presentation Capabilities

The enhancements described in this section have created a more efficient and accurate diagnostic presentation environment for the maintenance technician with a data validity check on test outcome information to control entry of incorrect or out-of-bounds test values and a feedback entry for the maintenance technician to indicate an unsuccessful maintenance action.

Data Validity Check

The maintenance technician must enter the results of a test. In previous versions, no test data validity checks were available to the technician to designate whether a particular test result was within the expected values of an acceptable test result. For instance, a particular voltage check (test) on a wire within a wire bundle should result in 5 +/- .01 volts for a pass and 0 to 4.989 volts for a fail, indicating the acceptable range of test voltage values is between 0 and 5.01 volts. In previous versions, if the maintenance technician tested the incorrect wire in the bundle and returned a value of 6 volts from the check, the options would be to pass or fail the test. After making a test entry, the presentation module would accept the word of the technician that the test was performed properly and the results were correct. This incorrect test entry was then provided to the diagnostic module. As a result, diagnostics proceeded down the wrong path to fault isolation and repair.

As a result of this investigation, the diagnostic presentation system was equipped with a data validity check, which retrieves pertinent test data values from the data base and requires a value from the maintenance technician that is within the expected realm of the test outcome.

Maintenance Action

Maintenance actions are rectifications that do not remove and replace (R&R) components. Rather they are some sort of in situ maintenance such as adjust, align, clean, etc. Previous versions of the presentation module treated each maintenance action as an R&R and did not consider instances when a repair or an R&R would be required if the maintenance action was unsuccessful or could not be performed properly. For instance, consider what would happen if a maintenance action on a component could not be performed successfully and the presentation module only acknowledges R&Rs. First, the presentation module would require a functional test even though nothing was fixed. Then, the diagnostic module, being given incorrect information on the outcome

of the maintenance action, could suggest another maintenance action on the same component, perform an R&R on another component ignoring the faulty one, or perform further unnecessary tests on the faulty component or other components.

A maintenance action, when used as a rectification, presents the unique situation of a passed test requiring a system health check. The reason is the maintenance action, if successful, was a rectification (with a system health mapped as its conformation test). But, if the maintenance action was unsuccessful it mimics a test that implicates a set of faults. An example, as illustrated in Table 7, is a system with potential faults of Out Of Alignment and Will Not Align among its set of manifested failures. The Out Of Alignment requires an alignment maintenance action while the Will Not Align requires an R&R. If the align rectification is accomplished, its success must be determined before proceeding. The test mapped to the Out Of Alignment is, "was the maintenance action successfully completed?" If the answer is yes, then the system health check must be performed to ensure the Out Of Alignment fault was the fault present in the system and the alignment did in fact remove the Out Of Alignment fault. If the answer is no, then the fault Will Not Align is implicated and diagnostics and repairs associated with that set must be performed. A third option occurs if the alignment was started but could not be completed. In this case, both the Out Of Alignment and Will Not Align faults are implicated and the repair actions required by these faults are indicated. Hence, each of these outcomes obviously produces different sets of implicated and exculpated faults and system health information.

Table 7. Maintenance Action Test Example

Select the output that represents the results of the maintenance action.			
	Completed?	Successful?	
Outcome #1	yes	yes	(pass)
Outcome #2	yes	no	(Fail 1)
Outcome #3	no	---	(Fail 2)
Outcome #1	Exculpated both faults -- Out of Alignment and Will Not Align		
Outcome #2	Implicates fault -- Will Not Align		
Outcome #3	Implicates both faults -- Out of Alignment and Will Not Align		
The maintenance action is mapped to Out of Alignment while the repair is mapped to both Out of Alignment and Will Not Align.			

Therefore, the logic to handle this unique situation requires data element modifications and presentation software modifications. Within the CDM rectification data elements for maintenance actions, the author is required to list both tests (aligned and system health) in sequence to prove that the Out Of Alignment fault was at fault and that the maintenance action did fix the problem. The first test is an MOT similar to the example below or a binary test. The second is the system health and is recommended if the first test passes.

VI. IMIS-DM DIAGNOSTIC AND USER INTERFACE FUNCTIONS

The IMIS-DM was combined with an automatic TO data presentation system and custom-designed user interface. This combination will provide a diagnostic session user interface for a test program to be accomplished on F/A-18 aircraft in mid-1991. The data base, presentation system, and IMIS-DM were installed on the PMA designed solely for the effective presentation of technical data. All experience to date in the human interface and data requirements for these presentations were combined in a specification for the human interface design of the system. Design requirements discovered during an earlier test at Homestead AFB, Florida, on F-16 aircraft were also included. The principal diagnostic-related capabilities of the combined system are outlined in the following paragraphs.

Manual/Automatic Symptom Loading

A hardware/software controller to activate and use the aircraft data bus was incorporated on the portable computer. Two forms of symptom loading (functional check result entries) were input back to the IMIS-DM for initialization: automatic and manual. Figure 9 represents the flow of information for automatic and manual feedback in IMIS-DM diagnostics.

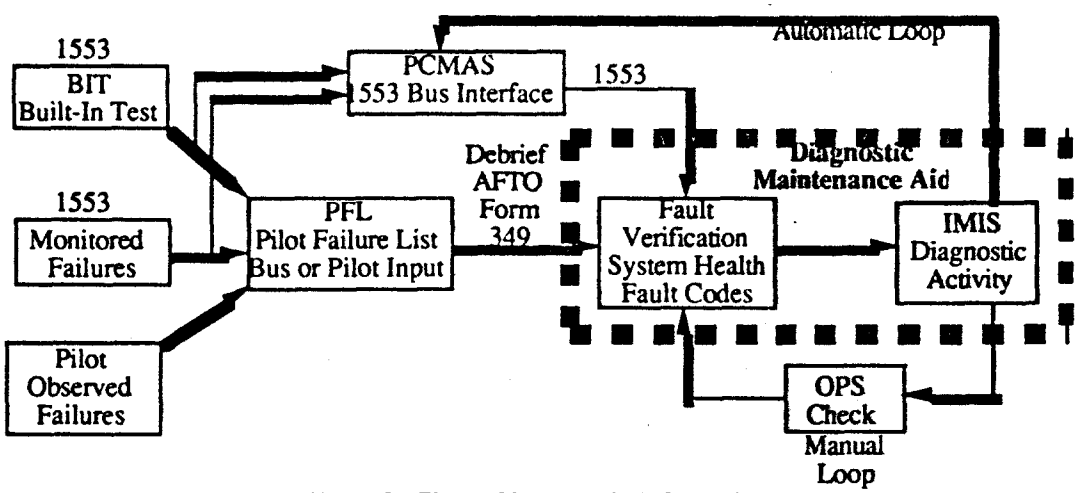


Figure 9. Flow of Diagnostic Information.

During initialization, the diagnostic module provides the maintenance technician with the opportunity to enter the fault/symptom information manually. Manual fault/symptom information was provided to the maintenance technician from pilot input data and previously performed MIL-STD-1553 data bus downloads. Upon completion of manual fault/symptom entry, automatic feedback through the MIL-STD-1553 data bus initializes BITs and returns symptoms directly to the IMIS-DM for symptom verification and entry.

Automatic Data Collection

Detailed maintenance data collection systems should provide the data needed to make the diagnostic aiding system really efficient. For example, data on test times, fault occurrence rates, access and closure times, and remove and replace times are needed. Data collected today, such as component failure rates and total task times, are accumulated by aircraft type, which tends to hide individual location variations. For example, an aircraft in a coastal environment may have corrosion problems with a particular component, while an aircraft of the same type at a landlocked

base may not. The diagnostic aiding system has been developed with the ability to collect, collate, and update all these pertinent parameters on a real-time basis. However, the facility to capture and store this information over an extended period of time for later use does not yet exist. When implemented for analysis, this capability will provide the technician with diagnostic data tailored to local peculiarities of environment and operations.

Log File

A utility to record major actions taken in a diagnostic sequence is implemented in the IMIS-DM. This utility is called the log file. The log file has numerous applications not only in the IMIS-DM but also in the general maintenance area. The ability to record a diagnostic sequence facilitates the re-creation of that diagnostic sequence at a later date for either review or training purposes. The diagnostic sequence in a particular log file may be examined side by side with other sequences to compare diagnostic paths. Furthermore, diagnostic sequences may be extracted from the log file to facilitate training activities as examples or exercises for students, and can also be analyzed for information concerning the supplies, equipment, manpower, and costs associated with specific diagnostic sequences, equipment, or operating locations.

The log file allows the IMIS-DM to be a more complete diagnostic tool by providing information about actions taken during a repair session. This information is intended to be analyzed out of the IMIS-DM. It has already shown its utility in the development of a feedback analysis tool which generates manpower, spares, support equipment, and other items of logistics concern.

The operation of the log file is fairly simple and straightforward. The log file is implemented using a major keystroke accumulator which saves the actions taken and the time required to complete those actions. At the end of a diagnostic session, the information is written to an external file that can be accessed outside the IMIS-DM.

Display Tests/Rectifications

During a diagnostic sequence, the technician may wish to view any options that are available to expedite or complete diagnostics. This is an important part of the man-machine interface of the diagnostic tool. Technicians must be able to incorporate their maintenance expertise in any given diagnostic sequence. A diagnostic aid which ignores operator expertise is not only inflexible but also impractical. In many cases, the technician progressing through diagnostics is able to come to conclusions about the problem due to sheer intuition or similar past experiences.

This concept was demonstrated by lists of all available rectifications and tests available to the technician. The feature enabled the technicians to evaluate all the possible options available to isolate or rectify a problem. Upon viewing these lists, the technician could evaluate the situation and either comply with the machine's recommendations or use personal expertise and experience to select a different option which would isolate or rectify the problem. Additionally, such lists were helpful in evaluating "what if" questions and enhancing the training capabilities of the tool.

Display Interleaved Tests/Rectifications

Studies completed during research of human interface issues revealed that technicians desire access to as much information as possible about the diagnostic problem on which they are working. Therefore, the need for ranked isolation and repair options resulted in the implementation of a display interleaved actions list. This enhancement to the diagnostic module data display was achieved by the interleaving of actions analysis described in Section IV. The technician selected the menu function of interleaved actions and a mixed hierarchical list of the top five actions provided

convenient viewing of the options that will best lead to fault rectification. The maintenance technician then had the opportunity to either select the recommended option or choose among alternative options presented in the list.

Review Previous Actions

Another function which enhanced flexibility of operation and gave the user more information is the review previous actions function. This facility was implemented to allow the technician to view all the tests and actions already accomplished in the diagnostic sequence. This feature is accessed via a function key. When called, it displayed to the user a complete ordered list of all tests and actions accomplished during the diagnostic session, along with the result of that activity. This type of function gave the technician information as to what was accomplished, which made for a more efficient diagnostic sequence by avoiding repeated actions. This feature is very useful when a technician must complete a diagnostic activity initiated by another or when the diagnostic activity has been suspended.

Table of Contents

In any interaction with TO data, eventually the user will want to choose a new point of entry into the data. Consequently, a table of contents facility was created to give the users an interface with a "look and feel" much like that they were familiar with in using Air Force TOs (AFTOs). The feature provided a bonus capability in diagnostics as well. In some cases, especially with immature data bases, there may arise occasions when the diagnostic module is simply unable to provide further assistance in fault isolation. In such a case, it is essential that all the technical data available to describe a system and prescribe repair actions be available to the technicians. At that point, they will be working on the basis of intuition and their own knowledge of the system, and it is imperative they have access to all available data. Hence, a table of contents facility was provided on the portable computer.

Menu Bar Operation

The graphic human interface developed for the IMIS-DM PMA has employed several improvements designed specifically to aid the maintenance technician in starting and completing a diagnostic session. The human interface is designed to a combination MOTIF and GCSFUI standard requirement as modified by AFHRL. The human interface employs a menu bar to allow the technician access to the data needed to perform the job effectively. The menu bar contains nine elements of which four contain items directly applicable to a diagnostic session: (a) status, (b) library, (c) troubleshooting, and (d) utilities.

Status

The status menu provides ready access to current information concerning the current state of the weapon system. An A/C profile provides configuration information used to select data that can be variable between aircraft and to alter the preferred diagnostics path. A maintenance profile provides information concerning the current maintenance status of the aircraft. Elements of this profile, such as doors opened and closed and power on and off, can change as the diagnostic session proceeds. A log file provides a continuous running history of all actions taken since the diagnostic session started along with results of those actions. A symptoms list shows a list of the symptoms encountered since the beginning of the diagnostic session. A test results listing shows all the tests run since the beginning of the session and their results. An ETIC selection initiates calculation based on current system state and shows ETIC for the most likely fault currently plausible and the probability that the system will have been repaired by expiration of that time.

Library

The library menu provides two elements very useful to the diagnostic process. A troubleshoot submenu allows the technician to select the mode of troubleshooting to be used during this session. Three mode selections are available. A troubleshooting tree selection allows the technician to proceed down a fixed tree of troubleshooting sequences. This selection is currently inoperative as data to support this tree have not been developed. A computer-assisted troubleshooting selection initiates the connection sequence to allow the technician to connect the IMIS-DM PMA to the MIL-STD-1553 data bus so that aircraft computers can be tied directly to the PMA during the diagnostic session. A manual troubleshooting selection initializes the IMIS-DM PMA to accept only technician inputs rather than computer inputs. The PMA is not connected to the MIL-STD-1553 bus; the technician acts as the PMA to system interface.

An aircraft history selection allows the PMA and the technician access to aircraft history needed to facilitate CND processing and to provide the technician additional information. This may allow him to make more intuitively correct selections of appropriate actions than can the IMIS-DM, which is based solely on the data contained in the current data base.

Troubleshooting

The troubleshooting menu provides the technician ready access to key places of diagnostic data at any point during a maintenance activity. The selections available are (a) ranked actions, (b) ranked tests, and (c) ranked repairs. These elements were explained above. They are provided on this menu so that they can be displayed any time the technician wishes to see them instead of just when the IMIS-DM is awaiting a task selection.

Utilities

The utilities menu provides access to a calculator and a digital multimeter. These two utilities are frequently needed during diagnostic sequences.

Diagnostic Graphic Interface

A diagnostic graphic interface is provided to help the technician visualize the diagnostic problem and observe progress as the fault isolation and repair process proceeds. The diagnostic graphic interface is a modified functional block diagram that shows fault connectivity rather than functional connectivity. Blocks in the diagram are items that can be rectified by the technician currently working on the system; shading of the blocks indicates the current state of the diagnostic session.

Diagnostic block diagrams are hierarchical; hence, at any functional or physical hierarchical level the block diagram can be used to indicate the current diagnostic state of all systems or components applicable to that level. As shading of the blocks in the diagram is essential to the intelligence provided by the diagram, all diagnostic block diagrams have very limited scrolling capability. Consequently, whatever diagram is visible on the screen will contain some degree of shading to denote the current state of the diagnostic session.

A sample screen is shown in Figure 10. A dark border indicates that potential faults in the item identified by the block are spanned by the currently recommended activity. A completely white box indicates a system element is not currently suspected of having anything to do with the reported symptoms. Dark shading indicates the item contains possible faults in the current plausible set. Light shading indicates an element contains possible faults temporarily removed from the plausible set. A diagonal line indicates all possible faults from the current symptom which might have been contained in that element have been exculpated as a result of successfully passing a test.

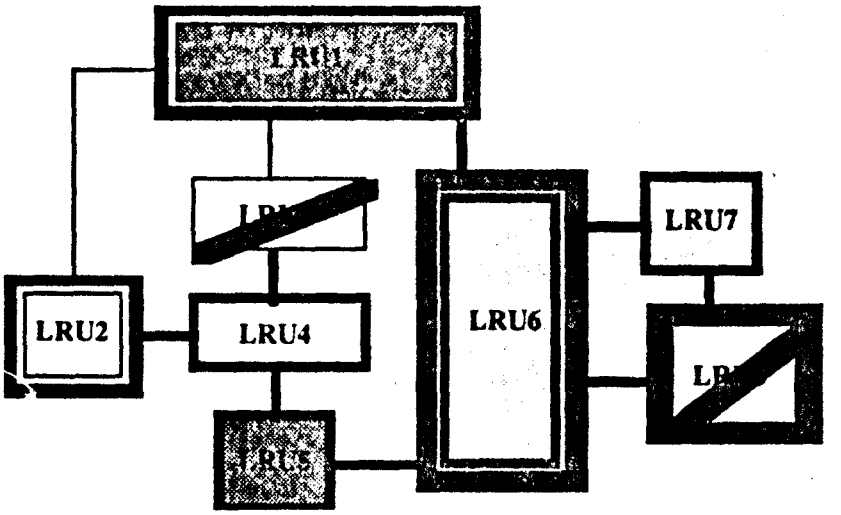


Figure 10. Diagnostic Block Diagram (Example).

Technical Data Presentation

An essential operating function of the IMIS-DM PMA is to provide the technician with all the data needed to complete the task assigned. Therefore, any time the technician selects a diagnostic activity, he is given the instructions for performing that activity. All technical instructions for the task at hand, whether opening a door, performing a test, interpreting test results, or performing a repair, are provided by the PMA upon selecting the task to be performed. Instructions provide both the text and the graphics needed to perform the job. This capability provides the integrated diagnostic information needed to complete the diagnostic task, provides a record of all diagnostic steps which had to be completed to task completion, and ensures the technician has available in a single source the full range of information needed to complete the assigned work.

VII. CONCLUSION

The IMIS-DM is the implementation of a powerful diagnostic strategy capable of handling multiple faults, MOTs, critical functions, and equipment availability. The strategy is founded on a fault-based approach which overcomes limitations of a component connection analysis yet avoids the needless detail of a low-level, bit-and-piece analysis.

The development of fault/component modeling techniques provided a flexible reachability matrix which computer evaluations could attack. This reachability matrix maps rectifications (components), tests, faults, and symptoms, providing the relationship needed for analysis, and creating a structure for repair of a fault. The reachability matrix also allows for the incorporation of fault probabilities and when expanded to include more than one symptom, demonstrated that the cause of a faulty system could be two or more independent faults which could be resolved by multiple fault evaluation.

Upon development of the reachability matrix, the theory of integrating fault isolation and rectification strategies provides the most direct route to fixing an aircraft with the least amount of time expended. A fault isolation strategy alone limits the steps taken to isolate a faulty component, but rectification of that component still needs to be performed. Including rectifications in the

diagnostic analyses allows a technician to rectify an aircraft system in the least amount of time by recommending actions that are so likely to solve the problem at hand that they are recommended prior to testing.

Other considerations had to be theorized and integrated into the diagnostic aid: at what cost and information gain would the performance of one selected activity outrank the performance of another, and what are the next step ramifications of each activity. This development involved the incorporation of time, unit cost, probability of occurrence, information gained, and forecasting of second step events.

REFERENCES

Cooke, G., Jernigan, J., Maiorana, N., and Myers, T. (1990a). Object-Oriented Integrated Maintenance Information System (IMIS) Diagnostic Module Development (AFHRL-TP-90-53). Brooks AFB, TX: Air Force Systems Command.

Cooke, G., Maiorana, N., and Myers, T. (1990b). Integrated Maintenance Information System (IMIS) Diagnostic Module Redesign (AFHRL-TR-90-XX). Brooks AFB, TX: Air Force Systems Command.

LIST OF ABBREVIATIONS

AFHRL	-	Air Force Human Resources Laboratory
AFHRL/LRC	-	Air Force Human Resources Laboratory/Combat Logistics Branch
AFSC	-	Air Force Systems Command
AFTO	-	Air Force Technical Order
ATOS	-	Automated Technical Order System
BIN	-	Binary
BIT	-	Built-In Test
BR	-	Best Rectification
CAD	-	Cartridge-Activated Device
CAEA	-	Complete and Enter All
CAEO	-	Complete and Enter One
CAMS	-	Core Automated Maintenance System
CDM	-	Content Data Model
CEMS	-	Comprehensive Engine Management System
CND	-	Can Not Duplicate
DC	-	Damage Code
DTD	-	Document Type Definition
EAFF	-	Exit at First Failure
ETIC	-	Estimated Time in Commission
F	-	Fault
FCK	-	Functional Check
FOD	-	Foreign Object Damage
FR	-	Failure Rate
HSD	-	Human Systems Division
IMIS-DM	-	Integrated Maintenance Information System Diagnostic Module
LRU	-	Line Replacement Unit
MIL-STD	-	Military Standard
MOT	-	Multiple Outcome Test
MTBF	-	Mean Time Between Failures
OO	-	Object-Oriented
PF_n	-	Probability of Fault n Occurring
PMA	-	Portable Maintenance Aid
PMC	-	Partially Mission Capable
POS	-	Probability of Success
PS	-	Plausible Set
R	-	Rectification
R&D	-	Research and Development
R&R	-	Remove and Replace
RIU	-	Remote Interface Unit
RRT	-	Removal and Replacement Time
RT	-	Rectification Time
S	-	Symptom
SEI	-	Systems Exploration, Inc.
SGML	-	Standard Generalized Markup Language
SRU	-	Shop Replacement Unit
T	-	Test
TO	-	Technical Order
TOC	-	Table of Contents
TT	-	Test Time
UFS	-	Units From Supply

GLOSSARY

Action. A diagnostic or corrective procedure performed by a maintenance technician.

Aircraft Configuration. Placements or layouts of aircraft system components.

Availability. A component's or test equipment's obtainability for use in the diagnostics process.

Best Rectification. A diagnostic software algorithm that chooses the optimum from among available rectification actions.

Best Test. A diagnostic software algorithm that chooses the optimum test from among those available at any point in the diagnostic sequence.

Component. The lowest physical level of indenture on which a maintenance technician at a given level of maintenance, (i.e., organizational, intermediate, and depot (O, I, or D)), will normally work. For example, an organizational level maintenance technician would consider a Line Replacement Unit (LRU) as a component; whereas, an intermediate level technician would consider the LRU an end item and the Shop Replacement Unit (SRU) a component.

Critical Rectification. A rectification that acts to repair all faults declared critical.

Critical Test. A test that examines all potential faults declared critical.

Criticality. A measure of need for a particular system capability. For example, a fault in an air-to-ground function might not be critical for an air defense sortie, whereas a fault in an air-to-air function would be critical for the same sortie requirement.

Dominant Action. A rectification action whose likelihood of success is so great that it is recommended before available tests that would reduce the plausible set.

Failure Rate. The inverse of Mean Time Between Failures (MTBF).

Fault. The manifestation, through either inference or direct observation, of a failure within a system.

Feedback Analysis. The process of collecting parameters while in the maintenance/diagnostic environment and using these parameters to update current logistics information.

Feedback Loop. An interconnection of faults and signals such that no single test point can successfully isolate the fault location.

Functional Check. A test performed to ensure that a rectification action has been successful in restoring a system to operational status.

Maintenance Action. A rectification that does not involve removal and replacement of a component, but is merely an adjustment.

Mean Time Between Failures (MTBF). The unit of reliability used in the IMIS-DM as a predictor of fault likelihood. Its inverse is the failure rate.

Multiple Faults. An event where two or more faults (failed components) exist simultaneously in a given system.

Multiple Outcome Test (MOT). A test procedure without a binary pass/fail result. The procedure may have any number of outcomes; however, each outcome is unique and distinguishable from all other outcomes.

Plausible Set. The set of possible faults that could logically have led to an observed or indicated faulty condition. The elements in this set of faults contain single faults or combinations of faults that are not redundant.

Rectification. The repair of a fault(s) which alleviates a symptom or set of symptoms.

Repair Time. The time required to complete system repair after a fault is isolated. It may include access times. It will include reinstallation of original components removed unnecessarily as part of diagnostics, secure and closure, and final functional check.

Support Equipment. Tools or devices needed to perform an action.

Test. A prescribed sequence of actions whose result will implicate or exonerate a set of faults.

Test Time. The time required to perform a test. It includes access time, time to gather necessary test equipment and tools, time to conduct the test procedures, and time needed to record/interpret test results.