

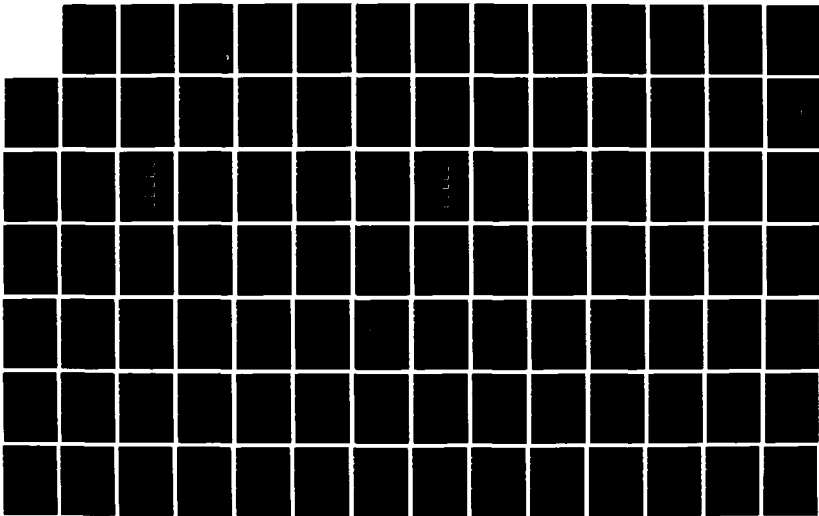
AD-A190 681

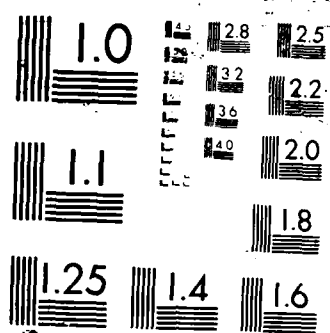
A SYSTEM STATUS MONITOR FOR THE NATIONAL AEROSPACE
PLANE(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
SCHOOL OF ENGINEERING J H BRAHMAN DEC 87
AFIT/8CE/ENG/87D-1

1/2

UNCLASSIFIED

F/G 1/3. 12 ML





AD-A190 681



DTIC FILE COPY

①

A System Status Monitor
for the National Aerospace Plane

THESIS

James M. Baumann
Captain, USAF

AFIT/GCE/ENG/87D-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

This document has been approved
for public release and sale; its
distribution is unlimited.

DTIC
ELECTE
MAR 25 1988
S E D

88 3 24 480

AFIT/GCE/ENG/87D-1

A System Status Monitor
for the National Aerospace Plane

THESIS

James M. Baumann
Captain, USAF

AFIT/GCE/ENG/87D-1

DTIC
ELECTE
MAR 25 1988
S E D

Approved for public release; distribution unlimited

A System Status Monitor for the National Aerospace Plane

THESIS

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

James M. Baumann, B.S.E.E.

Captain, USAF

December, 1987



Accession For	
NTIS GRA&I	☒
DTIC TAB	☒
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Preface

The purposes of this study were to develop a model for an in-flight system status monitor that could be applied to the National Aerospace Plane, and to implement a computer program to demonstrate the feasibility of that model.

The system status monitor model which I developed features dual hierarchical structures, one for the aircraft components and functions to be diagnosed, and another for the diagnostic functions to be performed. The aircraft knowledge base included elements from each level of the aircraft hierarchy, from primitive components through the overall mission. The diagnosis hierarchy which was implemented only included diagnosis and remediation. The addition of the other diagnostic functions to the demonstration program would be a valuable project for future students.

I wish to thank several people for helping with this thesis effort. My thesis advisor, LtCol Charlie Bisbee, invariably offered suitably probing questions and subtle guidance. Ms Kathy Abbott provided me with a copy of her Faultfinder software and considerable insight into its theory and operation. Capt Carl Lizza steered me in Kathy Abbott's direction, thus making implementation of my ideas much easier. Mr Mike Snead offered enthusiasm and support when others were skeptical. Finally, I offer special thanks to my wife, Christine, who did the work of three for our family, instead of her usual two, while I was doing other things.

James M. Baumann

Table of Contents

	Page
Preface	ii
Table of Contents	iii
List of Figures	vi
Abstract	viii
I. Introduction	1
1.1 Problem	1
1.2 Scope	2
1.3 Assumptions	2
1.4 General Approach	3
1.5 Sequence of Presentation	4
II. Problem Analysis	6
2.1 Problem Definition	6
2.1.1 NASP Domain	6
2.1.2 Status Monitor Functions	7
2.2 Literature Review	8
2.2.1 Automated Medical Diagnosis	8
2.2.2 Automated Hardware Diagnosis	11
2.2.3 Hardware Diagnosis in the Flight Domain	12
2.2.4 Trends in Automated Diagnosis	13
2.2.5 Blackboard Systems	13

	Page
III. Theoretical Development	15
3.1 Functional and Diagnostic Hierarchies	15
3.1.1 Functional Hierarchy	15
3.1.2 Diagnostic Hierarchy	18
3.2 Semantic Network Knowledge Representation	23
3.3 Causal Knowledge and Reasoning	25
IV. Program Development	32
4.1 Possible Programming Approaches	32
4.1.1 Expert System Shells	32
4.1.2 Blackboard Shells	33
4.1.3 Faultfinder	33
4.2 NASP Knowledge Base	35
4.2.1 NASP-specific Knowledge	35
4.2.2 Extending the Knowledge Base	36
4.3 Monitoring Function	37
4.4 Diagnosis Function	37
4.4.1 Stage 1	39
4.4.2 Stage 2	39
4.4.3 Diagnosis Displays	42
4.5 Remediation Function	47
V. Results	58
5.1 Test 1A	58
5.2 Test 1B	62
5.3 Test 1C	68
5.4 Test 1D	68
5.5 Test 1E	71

	Page
5.6 Test 2A	73
5.7 Test 2B	77
5.8 Test 2C	77
5.9 Test 2D	80
5.10 Test 2E	80
VI. Conclusions and Recommendations	85
6.1 Conclusions	85
6.2 Recommendations	85
A. Explanation of the NASP Aircraft Systems	88
A.1 Propulsion System	88
A.2 Hydraulic System	89
A.3 Fuel System	89
A.4 Flight Control System	89
A.5 Thermal Protection System	89
B. Listing of the NASP System Status Monitor Knowledge Base	90
C. Users' Manual for the NASP System Status Monitor	112
C.1 File Structure	112
C.2 Installation	112
C.3 Operation	113
C.3.1 Loading	113
C.3.2 Start-up	113
C.3.3 Using the SSM	113
Bibliography	115
Vita	117

List of Figures

Figure	Page
1. Functional Hierarchy.	16
2. Expansion of the Functional Hierarchy.	17
3. Diagnosis Hierarchy.	19
4. Steps in the Diagnostic Process.	24
5. Object Classes in the Semantic Network Knowledge Base and the Object Instances in Each Class.	26
6. Semantic Network Relationships.	30
7. Systems, Sensors, and Sensor Values.	38
8. Sample Fault Propagation Tree.	40
9. NASP SSM System Display.	43
10. Sample Fault Hypothesis Listing.	45
11. Functional Hierarchy Levels and Their Displays.	46
12. Mission Display.	47
13. Takeoff Display.	48
14. Climb Display.	48
15. Cruise Display.	49
16. Descent Display.	49
17. Landing Display.	50
18. NASP Aircraft Display.	50
19. Propulsion System Display.	51
20. Left Engine Display.	51
21. Right Engine Display.	52
22. Hydraulic System Display.	52
23. Left Hydraulic Subsystem Display.	53
24. Right Hydraulic Subsystem Display.	53

Figure	Page
25. Fuel System Display.	54
26. Flight Control System Display.	54
27. Thermal Protection System Display.	55
28. Sample Remedy List.	57
29. Test 1A Fault Tree Diagram (first hypothesis).	60
30. Test 1A Fault Tree Diagram (second hypothesis).	61
31. Test 1A Remedies.	62
32. Test 1B Fault Tree Diagram.	64
33. Test 1B Remedies.	65
34. Test 1C Fault Tree Diagram.	69
35. Test 1D Fault Tree Diagram.	70
36. Test 1D Remedy.	71
37. Test 1E Fault Tree Diagram.	72
38. Test 2A Fault Tree Diagram (first hypothesis).	74
39. Test 2A Fault Tree Diagram (second hypothesis).	75
40. Test 2A Fault Tree Diagram (third hypothesis).	76
41. Test 2A Remedy.	77
42. Test 2B Fault Tree Diagram.	78
43. Test 2C Diagnosis.	79
44. Test 2D Fault Tree Diagram (first hypothesis).	81
45. Test 2D Fault Tree Diagram (second hypothesis).	82
46. Test 2D Fault Tree Diagram (third hypothesis).	83
47. Test 2E Fault Tree Diagram.	84

Abstract

The purposes of this study were to develop a model for an in-flight diagnostic system that could be applied to the National Aerospace Plane, and to implement a computer program to demonstrate the feasibility of that model as a basis for a system status monitor.

The diagnostic system model which was developed features a double hierarchy structure, one for the aircraft functions to be diagnosed, and another for the diagnostic functions to be performed. The hierarchical nature of both the system knowledge and the functions that use the knowledge allow decomposition of the diagnostic task into relatively independent and manageable parts.

The demonstration program which was developed includes a subset of the diagnostic system model. This program was implemented in Zetalisp on a Symbolics 3600 computer. It will simulate monitoring the dynamic performance parameters of an aircraft's subsystems, report any readings that fall outside of predetermined limits, reason about components responsible for the fault, display to the aircrew the other aircraft functions which may be affected by the component fault, and recommend actions that may remedy the fault situation.

The demonstration program clearly shows the validity of the diagnostic system model and highlights the importance of the causal and functional relationship techniques used to represent knowledge of the aircraft and its environment. The program demonstrates how the diagnostic system can supply relevant system status information to the aircrew. The report concludes with several recommendations for enhancements to the demonstration program.

A System Status Monitor for the National Aerospace Plane

I. Introduction

A general and continuing trend in aerospace vehicles is their increasing complexity. These vehicles are becoming larger, are operating at higher altitudes and greater speeds, and are expected to perform with greater reliability. Despite this rapid increase in the complexity of aircraft, the crewmembers who operate them must use human decision-making capabilities which have remained relatively constant over the years.

Perhaps the extreme example of a complex aerospace vehicle is the proposed National Aerospace Plane (NASP). The NASP will be able to take off from a conventional runway, and either cruise at hypersonic speeds in the upper atmosphere, or accelerate to speeds sufficient to attain low earth orbit. The prototype NASP aircraft, designated the X-30, will demonstrate this mission capability with as few as two or three crewmembers [22]. Such a complex vehicle, performing such a demanding mission with a minimum crew, will require extremely well-designed aids to help the aircrew maintain full control of the aircraft. The aircrew aids will be especially important if and when abnormal conditions arise in-flight.

1.1 Problem

The problem investigated in this study is to develop and demonstrate a strategy for an in-flight system status monitor for the National Aerospace Plane. This monitor should be able to assess the health and status of various aircraft systems, recognize deviations from normal operation, diagnose the causes of the faults, and report the possible consequences of the faults to the aircrew. Because of the complexity of the NASP, the system status monitor strategy must account for the intricate interaction of aircraft systems. The system status monitor should help increase the decision-making capabilities of the aircrew so they can keep pace with increasing aircraft complexity.

1.2 Scope

While the system status monitor developed in this report can, in theory, include every aircraft subsystem and component, the demonstration program only includes five major aircraft systems; propulsion, fuel, hydraulics, flight controls, and thermal protection. The demonstration program also fully implements only the diagnosis and remediation aspects of the full diagnostic process. This set of aircraft systems was considered sufficient to investigate the interactions between and within systems. Also, the two diagnostic functions were enough to show the feasibility of using artificial intelligence techniques to perform system status monitoring.

1.3 Assumptions

In this study, all discussions of the diagnostic process will assume that single faults cause all observed fault symptoms at any single point in time.

Since the National Aerospace Plane is still in the planning stages, all references in this study to its missions, capabilities, and configuration are based on conjecture. One example of this is the model of the NASP propulsion system used in this study. While the actual NASP may use any one of a variety of propulsion plants, the one modeled here is the airturbo ramjet (ATR) [21]. With a maximum speed capability of about Mach 6, the ATR would not be sufficient by itself to propel the NASP to orbital speeds, but it could be used in conjunction with other propulsion technologies.

While the actual NASP will certainly require more, only five aircraft systems are modeled here. They are propulsion, fuel, hydraulics, flight controls, and thermal protection. These were considered the most important, and should be sufficient to illustrate the concepts of NASP system status monitoring.

These assumptions had some effect on the quality and completeness of the diagnostic knowledge base. The causal or functional knowledge, which is based on the defined structure of the aircraft, was not greatly affected because it corresponded to the aircraft as it was artificially de-

fined for this study. However, the compiled or experiential knowledge was very sparse because there was no actual experience with this aircraft to draw upon. The final results of the study are still valid because the representative cases used to test the demonstration program show the effective use of both types of knowledge.

1.4 General Approach

This study was undertaken as a series of steps leading from research into the nature of the NASP mission and the diagnostic process to the development and testing of a computer program which demonstrated the feasibility of the diagnostic model. The research into the nature of diagnosis showed that the process of diagnosis actually is at least a two-step activity involving system monitoring and then fault isolation. To become more useful for aircraft system status monitoring, diagnosis can be extended to a five-step process, as will be discussed in Chapter III.

To implement the multi-step diagnosis model, different artificial intelligence problem-solving techniques were investigated. The most promising found was the blackboard problem-solving model. A blackboard is a structured, global database which serves as a central repository of information to be accessed by separate and independent expert systems [8, 3]. Blackboards and their application to the NASP system status monitoring task will be discussed at length in Chapter IV.

The research next turned to a search for a suitable expert system shell that could support the blackboard model. Several general purpose shells were found, but all were either themselves in development or were not readily available at this Institute. A special purpose aircraft diagnosis system was found in development in the Vehicle Operations Branch of the NASA Langley Research Center. This system, called Faultfinder, uses a blackboard data structure to organize interaction between the different parts of the program. Faultfinder became the basis for the NASP system status monitor reported here.

Prototype development involved a number of modifications and extensions to the Faultfinder

system. Faultfinder's target domain is commercial transport aircraft, and its knowledge base and user interface were developed for that domain [19, 1]. The first task was to adapt Faultfinder to the NASP domain. Next, Faultfinder was modified to perform diagnosis on multiple levels of the aircraft functional hierarchy. Finally, a remediation function was added to propose actions that could be taken by the aircrew given a certain fault diagnosis.

Finally, the modified Faultfinder system was tested with several sets of theoretical fault symptoms. The system performed adequately in most cases, but a number of areas needing improvement were discovered. Recommendations are made in Chapter VI as to the implementation of these improvements.

1.5 Sequence of Presentation

Analysis of the problem of system status monitoring for the NASP is presented in Chapter II. This includes definition of the problem, and a review of the literature related to diagnosis.

Chapter III covers the theoretical development of the system status monitor model. Here, the diagnostic and aircraft functional hierarchies are developed, knowledge representation issues are discussed, and methods of diagnostic reasoning are explored.

Development of the system status monitor demonstration program is presented in Chapter IV. First, the potential solution approaches are compared, and the reasons for choosing Faultfinder as the basis for the NASP System Status Monitor are explained. Next, the task of transforming Faultfinder into the NASP SSM is described. This description includes the representation of the aircraft's physical and functional interrelationships, the format of the status monitor displays, and logic of the diagnosis and remediation algorithms.

In Chapter V, performance of the prototype system status monitor is discussed. This discussion includes results of test runs using simulated fault symptom inputs. Finally, Chapter VI

provides conclusions which can be drawn from this study and recommendations for further work in this area.

II. Problem Analysis

2.1 Problem Definition

The problem to be addressed in this study can be divided into two related issues: 1) Why does the National Aerospace Plane need a system status monitor, and 2) What should the system status monitor do?

2.1.1 NASP Domain The National Aerospace Plane (NASP) will be a revolutionary transportation system, capable of taking-off and landing horizontally on a conventional runway and ascending directly into orbit or cruising at 6 to 12 times the speed of sound at altitudes greater than 100,000 feet [22].

To perform its intended mission, the NASP must be extremely efficient, requiring some or all of its subsystems to perform multiple tasks. Examples of multi-purpose subsystems are the fuel system, where the cryogenic fuel may be circulated through hot structures to provide active cooling, and the forward fuselage, which may also serve as part of the engine inlet structure.

This interdependency of the aircraft systems will complicate the aircrew's normal system monitoring task. The effects of a fault in a particular system will probably not stay within that system, but will propagate to other systems. As aircraft systems become more complex and interdependent, the possible ramifications of any single fault on other aircraft systems become more complex and more difficult to trace.

The extremely large operational flight envelope of the NASP places added demands on the flight crew in two ways. Operation in one flight phase, such as takeoff, may require the aircraft systems to perform in much different ways than in another flight phase, such as hypersonic cruise. A fault within a system may not greatly affect the current flight phase, but may preclude successful completion of a later flight phase. These interrelationships must all be considered when assessing the status of the aircraft.

The other area where the large flight envelope of the NASP comes into play is real-time ground-based support. In the past, manned space vehicles such as Mercury, Gemini, Apollo, and the Space Shuttle have had extensive system monitoring support by personnel and equipment on the ground. This ground-based support was realized through worldwide communications networks. The NASP may not have the luxury of this extensive ground-based support, and therefore an on-board system status monitoring capability may be required.

System complexity, interdependence, the large flight envelope, and the requirement for autonomous operations, along with the speed with which events occur during hypersonic flight, will combine to dictate the automation of NASP system status monitoring.

2.1.2 Status Monitor Functions Once the need for an automated system status monitor has been established, the form and function of the status monitor must be defined.

As the name implies, a system status monitor should keep the flight crew apprised of the status of the aircraft systems. The monitor will need to keep track of the state of sensors which measure various aircraft parameters. If any sensor reports an abnormal reading, the monitor should diagnose the cause of the abnormality. While monitoring is a straightforward process, diagnosis can be a very difficult task when applied to even a moderately complex mechanical system. The collective processes of monitoring and diagnosis traditionally have been simply called diagnosis. Chapter III will discuss how this two step diagnostic process can be extended to provide additional information for the flight crew.

The complexity and interdependence of the NASP systems would further imply that the status monitoring task cannot be applied to each individual system as if it were operating alone. A NASP system status monitor will need to operate in the context of the aircraft as a collection of closely coupled, tightly knit systems.

The particular problem this study will address is the design and implementation of a system status monitor that can perform extended diagnostic functions on a complex aircraft with highly

interdependent systems.

2.2 Literature Review

Since the diagnostic process forms the basis for system status monitoring, this literature review will concentrate on diagnosis. The literature related to the blackboard problem-solving model will also be reviewed.

Diagnosis is usually defined in medical terms as "the act or process of identifying or determining the nature of a disease through examination [12, 363]." In recent years, the meaning and application of diagnosis have been expanded to include the domain of mechanical and electrical devices. In this context, diagnosis can be defined as the use of "situation descriptions, behavior characteristics, or knowledge about component design to infer probable causes of system malfunctions" [23, 34].

In both the medical and engineering fields, diagnosis has traditionally been a manual effort performed by a human expert in that field. To improve the quality of diagnosis in the medical field, and to cope with increasingly complex systems in the engineering field, researchers are currently investigating automated diagnostic tools. These automated tools usually take the form of "expert systems." While manual diagnosis forms the basis for most of the theory of diagnosis, this review will concentrate primarily on the current research into automated diagnosis.

2.2.1 Automated Medical Diagnosis One of the first and best-known medical diagnostic expert systems is MYCIN. It was designed to diagnose infectious blood diseases and to help the physician select the correct type and dosage of a drug treatment. MYCIN is a rule-based system that uses a backward chaining technique to reason from the patient's observed condition (the symptoms) to the identity of the infecting organism (the cause). The system was developed at Stanford University, and work on this project by Shortliffe, Axline, Buchanan, Merigan, and Cohen

was reported in the literature as early as 1973 [20]. MYCIN has served as a model or inspiration for several other medical diagnostic expert systems, including EMYCIN and NEOMYCIN [23, 326].

The DIALOG (for DIAGnostic LOGic) system, reported by Pople, Myers, and Miller [17], takes a more sophisticated approach to the medical diagnosis problem. It was designed to imitate the data structures and diagnostic reasoning processes used by a knowledgeable internist. The DIALOG system was able to correctly diagnose multiple (related or unrelated) diseases in the same patient. DIALOG demonstrated accurate diagnostic performance in cases involving as many as five distinct diseases [17, 848].

Rather than to simply search through a state-space as did MYCIN, DIALOG developed hypotheses about the causes of observed patient symptoms, and partitioned those hypotheses into disjoint sets. A form of deductive inference called abduction was then used to sequentially step through the sets of hypotheses, accepting the correct hypotheses and reject the incorrect hypotheses. The method of abduction required that DIALOG have control structures to deal with the following four issues:

1. Observations must be able to 'trigger' or evoke hypotheses of disease entities with which they are associated.
2. Hypotheses must be able to generate expectations concerning likely consequences, which may be posed as questions regarding additional observations . . . in order to 'test' the hypotheses.
3. It is necessary to provide some means for deciding among contending hypotheses.
4. Some means must be developed to group hypotheses into mutually exclusive subsets corresponding to coherent problem areas. [17, 849]

DIALOG's data structures consisted of three primary relationships to represent dependencies inherent in the internal medicine problem domain. These were

1. Manifestation (M) evokes disease (D),
2. Disease (D) is manifested by manifestation (M), and
3. One disease (D1) is a form of another disease (D1)

These relationships were organized into a network to represent the hierarchy of diseases and the associated manifestations. In the network, the diseases were represented as nodes, and each of the three relationships above were represented as directed arcs connecting the nodes. The network also contained two different weighting factors:

1. The likelihood that a certain disease is the cause of a particular manifestation, and
2. The frequency with which a patient with a particular disease will display a certain manifestation.

The weighting factors allowed DIALOG to choose the most likely of two or more competing hypothetical diagnoses [17, 849-850].

Pople expanded on his work with DIALOG by developing the INTERNIST-II medical diagnostic system [16]. Its area of expertise was also internal medicine. INTERNIST-II was one of the most extensive medical diagnostic expert systems, with descriptions of more than 500 separate diseases and more than 3500 disease manifestations [23, 281].

INTERNIST-II's major improvement over DIALOG was its ability to simultaneously view the disjoint sets of hypotheses and reason over the entire group of sets. This allowed it to more quickly converge on a correct diagnosis and, in some cases, yield a more accurate result [16, 1030].

In addition to the hierarchical networks of diseases and their manifestations used by DIALOG and INTERNIST-II, the next generation of medical diagnostic expert systems also incorporated models of general human physiological knowledge and specific information about the patient's physical state. One such system was ABEL [15, 893], which aided in the diagnosis of electrolyte and acid-base disturbances.

ABEL incorporated two unique features which set it apart from DIALOG and INTERNIST-II. First, it used its general and specific physiological knowledge to define what was called the "patient-specific model." Next, this model was used to construct and refine a multi-level network

that represented not just the simple associations between disorders and their manifestations, but also their causal relationships. It is this model of the patient's current state of health and the understanding of the cause of deviations from a normal state of health that make ABEL more sophisticated than its predecessors [15].

MDX was a medical diagnosis system developed at Ohio State University by B. Chandrasekaran and others. Its primary domain was a liver syndrome called cholestasis. MDX used a hierarchical knowledge organization which was operated upon by a collection of "cooperating experts." These experts communicated with each other through a blackboard structure [1]. This type of knowledge-based diagnosis system is very similar to the conceptual structure of the proposed NASP system status monitor.

2.2.2 Automated Hardware Diagnosis Expert systems for the diagnosis of electronic hardware faults are descendants of the medical diagnostic systems mentioned in the preceding section. As such, they have benefited from the evolution of the medical diagnostic systems and incorporate the most sophisticated and powerful features of the medical systems. As one might expect, the study of the hardware diagnostic process has led to the discovery of new and better ways to perform the general diagnostic task.

An example of the current work in expert systems for hardware diagnosis is reported by Randall Davis of the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology [3]. Davis embraces the idea that a diagnostic system will benefit from a causal understanding of the structure and function of the malfunctioning device in question. He adds two principles to the theory of diagnosis. These principles are layering the paths of interaction and the concept of locality [3, 88].

Davis represents the function and structure of the device as paths of causal interaction. These paths define the possible interactions between any pair of the device's components. In normal operation, there are a certain number of probable interactions between components, and an additional

number of possible but much less likely interactions. When also considering fault situations, the number of possible interactions becomes much larger. The paths of interaction are used to generate candidates to determine which components are causing a fault. If every conceivable interaction path is considered, the number of candidates to examine will become unwieldy. Conversely, if the number of interaction paths is too restricted, some entire classes of candidates may never be considered.

Layering the paths of interaction is used as a compromise between having too many and too few interaction paths. Each layer uses a different set of interaction paths to represent a different model of the device. By layering the models, the most restrictive model is considered first, and a less restrictive model is considered only if the first yields a contradiction [3, 90-93].

The concept of locality holds that the most appropriate representation of the malfunctioning device will be the one in which the cause and symptom of the fault are adjacent, or "local" to each other. Therefore, an electrical adjacency representation would be appropriate for a continuity fault, while a thermal adjacency representation would be more appropriate for a fault caused by heating [3, 94].

2.2.3 Hardware Diagnosis in the Flight Domain Hardware diagnostic systems being developed for application in the flight domain have incorporated most of the desirable features of the diagnostic systems examined so far and have also been enhanced with new and innovative capabilities. Much of this work is being performed by the National Aeronautics and Space Administration (NASA) at the Dryden Flight Research Facility, Edwards, California, and the Langley Research Center, Langley, Virginia [3,19].

The research being performed in the area of expert system fault diagnosis at the Dryden Facility is in support of the development of advanced digital flight control systems. The system reported in [4], called the experimental expert system flight status monitor (EESFSM), will be used by a flight systems engineer on the ground to assess the status of the flight control system

in a remotely piloted vehicle. The EESFSM goes beyond the diagnostic systems reviewed thus far in that it incorporates functions that are performed both before and after the actual diagnostic process. Before diagnosis is performed, the EESFSM uses expert system knowledge to monitor the fault status indicator bits of the flight control system to detect the presence of a fault symptom. The detected fault symptom then feeds the fault diagnosis capability. The EESFSM uses the results of the fault diagnosis to recommend corrective actions and to deduce situations of concern [4, 3].

In their research [19], Schutte and Abbott used a fault monitoring and diagnosis system similar to the EESFSM, but they extended its scope beyond the flight control system. They have developed a hierarchy of aircraft "goals," with the proper functioning of a subsystem being the lowest level of the hierarchy. A group of subsystems makes up the flight control system, which in turn is one factor contributing to the flight dynamics of the aircraft. The next two levels in the hierarchy are the aircraft trajectory and route, in that order [19, 3]. When the diagnostics function of this system identifies a fault, it can also determine the effect of that fault on the accomplishment of the other goals in the hierarchy.

2.2.4 Trends in Automated Diagnosis Automated diagnostic systems began as expert systems, with each fault situation represented in a separate rule. As this literature review suggests, the trend in automated diagnosis is toward a deeper representation of system knowledge. The deeper knowledge generally represents the normal operation and interaction of the system rather than specific fault situations.

2.2.5 Blackboard Systems The blackboard problem-solving model was first used in the HEARSAY-II speech understanding system developed in the early 1970's by Erman and others [5]. Since then, blackboards have been used in a wide variety of applications, and each time in a slightly different form [8, 2].

A blackboard architecture refers to a fairly simple concept that has been tailored to meet

the specific needs of its users. In its simplest form, a blackboard is a central database that can be accessed by independent program modules. These modules are called knowledge sources, and usually take the form of expert systems. One of the knowledge sources usually acts as the controller to determine which knowledge source will be permitted to have access to the blackboard next. The blackboard serves as the only means for the knowledge sources to communicate. If a knowledge source needs information, it looks for it in the blackboard. If a knowledge source can supply information, it posts that information to the blackboard for all other knowledge sources to see. In this way, the blackboard model supports incremental, opportunistic problem solving. Each knowledge source contributes its own small part of the problem solution, and does it only when its necessary inputs have appeared in the blackboard.

The interested reader can obtain a more detailed analysis of blackboard theory and applications from the excellent papers by Hayes-Roth [9,10,8] and Nii [13,14].

III. Theoretical Development

Development of the theory underlying the National Aerospace Plane system status monitor will be covered in three sections in this chapter. This discussion will center on a) the diagnostic and functional hierarchies which form the framework of the system status monitor, b) the semantic network form of knowledge representation used here, and its advantages versus an associational form of knowledge representation, and c) the causal knowledge representation and reasoning method used in the remediation level of the system status monitor. Specifics about how this theory was applied to the implementation of the NASP system status monitor will be covered in Chapter IV.

3.1 Functional and Diagnostic Hierarchies

As stated in the previous chapter, the complexity of the NASP will require that the system status monitor provide as much useful information as possible to aid the flight crew. The diagnostic and functional hierarchies defined here serve as a framework for providing that information to the flight crew. The functional hierarchy will be examined first.

3.1.1 Functional Hierarchy The functional hierarchy, shown in Figure 1, was derived from the goal hierarchy developed by Schutte and Abbott [19], which in turn developed from the work of Chen [2]. From top to bottom, each level in the functional hierarchy is composed of one or more instances of the level below it. Thus, the mission is composed of one or more flight phases, each flight phase has an instance of the aircraft to perform it, and so on. This expansion of the hierarchy is shown in Figure 2. An important point is that each flight phase has a different instance of the aircraft because different capabilities of the aircraft are required to accomplish each flight phase. Likewise, each aircraft instance has its own instances of each of the individual aircraft systems. This hierarchical framework helps to organize the knowledge about the aircraft and its functions. Any component or function at any level of the hierarchy can be associated easily with

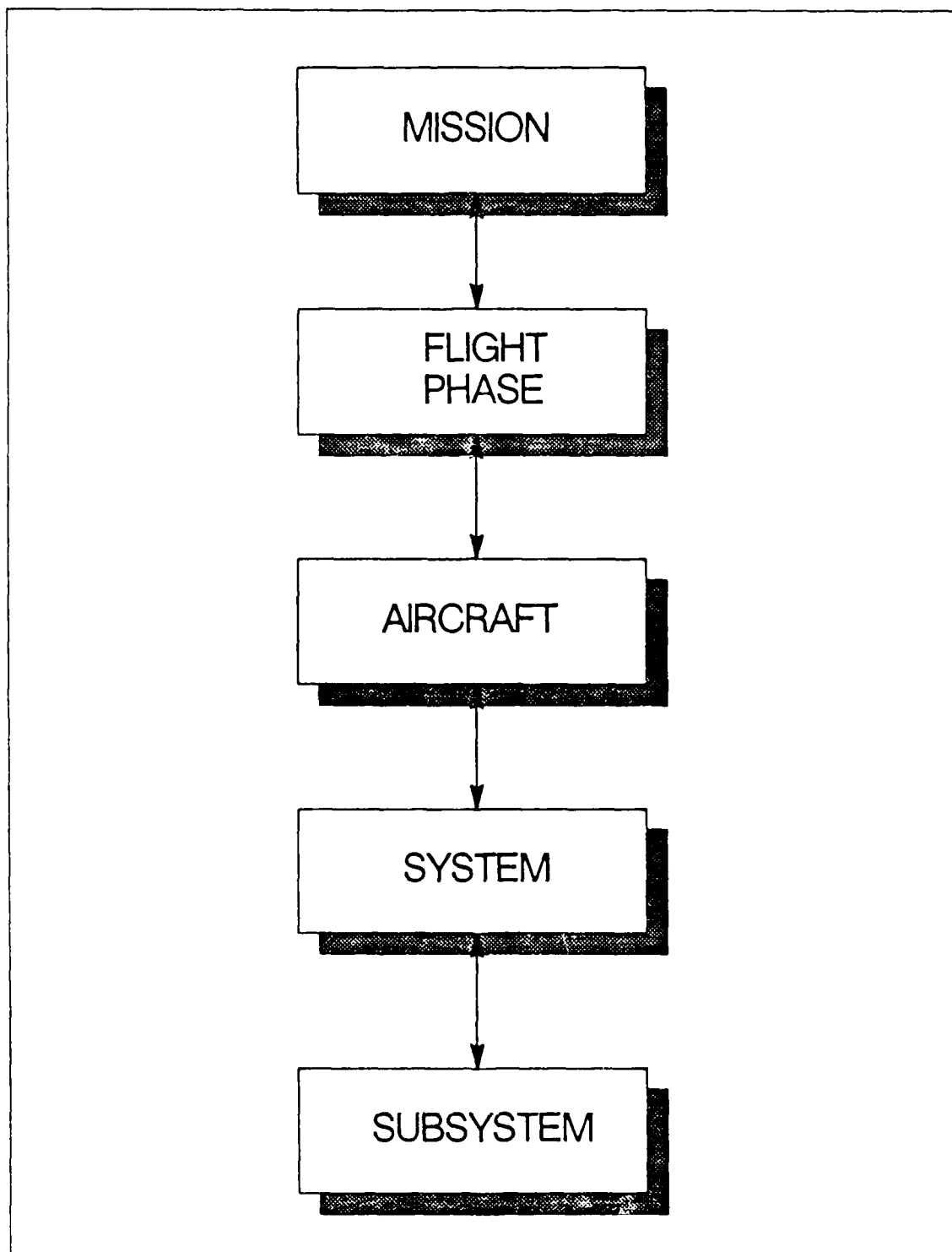


Figure 1. Functional Hierarchy.

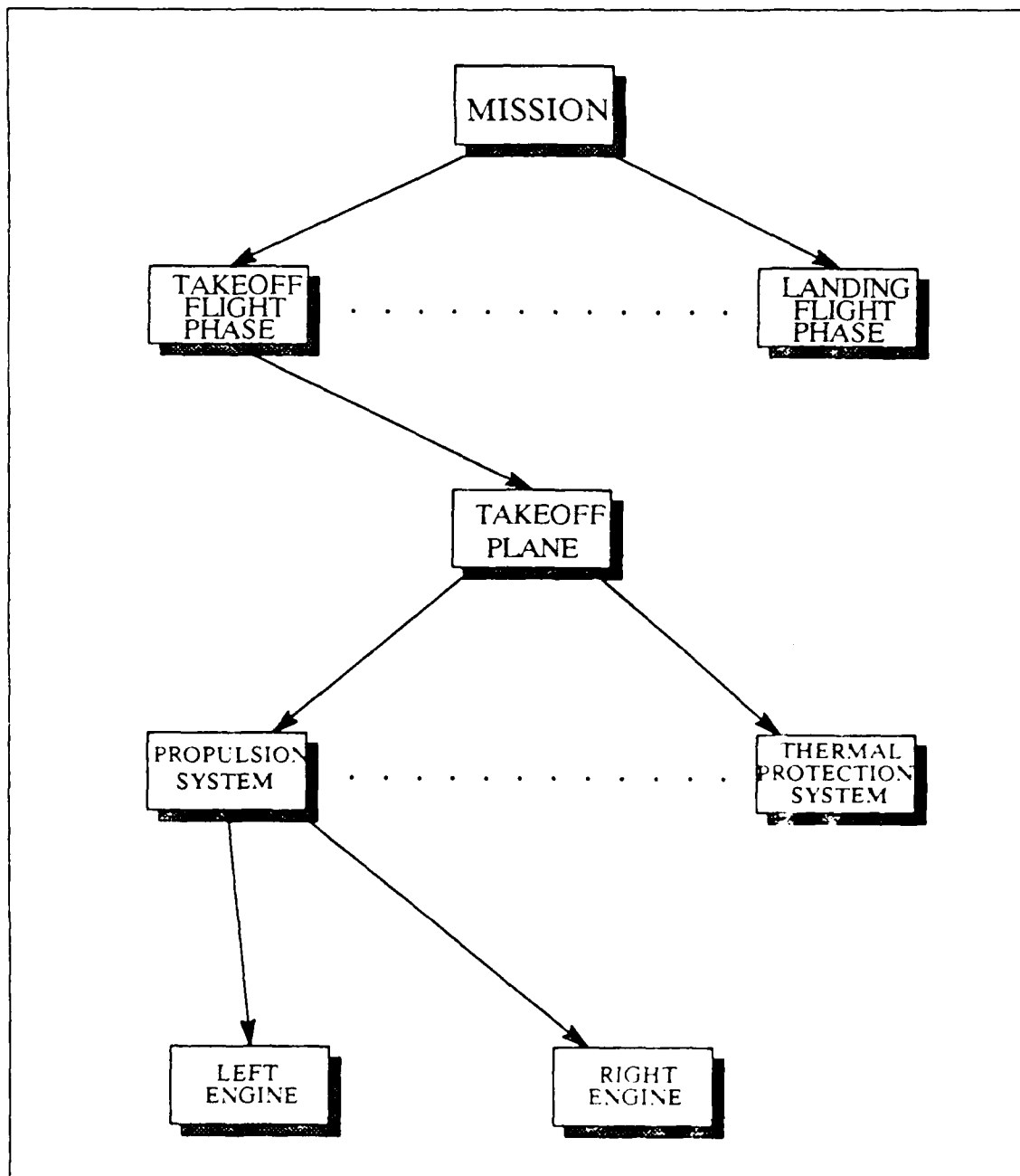


Figure 2. Expansion of the Functional Hierarchy

the components on which it depends (lower levels in the hierarchy) and also with the components that are dependent on it (higher levels in the hierarchy).

3.1.2 Diagnostic Hierarchy While this hierarchy is named for diagnosis, the actual diagnosis function is only one of five levels in the hierarchy. Figure 3 shows the diagnosis hierarchy and the relative positions of the five levels. To avoid confusion, the collection of all five levels will be called the "diagnostic process," and the second level of the diagnostic process will be called the "diagnosis function." The entire diagnostic process is performed bottom-up, with each level supplying its output information as input to the next higher level.

The definition of diagnosis used in the previous chapter was "to infer probable causes of system malfunctions." This definition implies that there be a method for determining if a system malfunction indeed occurred. This is the function of the first diagnostic level, monitoring.

3.1.2.1 Monitoring The overall diagnostic process is started by monitoring the physical system in question. The monitor must be able to detect a fault condition and report it to the next level in the diagnostic hierarchy. To do this, the monitor must first be able to discriminate fault conditions from normal conditions. Since normal operating conditions are usually understood better than fault conditions, the monitor usually starts with a model of the normal operation of the physical system. This model takes the form of a numerical simulation of the operation of the physical system. Readings from sensors in the physical system are compared to values that are predicted by the numerical simulation. If the sensed values fall outside of a range of acceptable predicted values, then a fault has occurred and it is reported.

In the case of the NASP, the fault monitor must contain numerical simulations that also account for the different flight phases. As an example, the model of the engines must predict a different range of normal readings for the takeoff phase than it would for the hypersonic cruise phase.

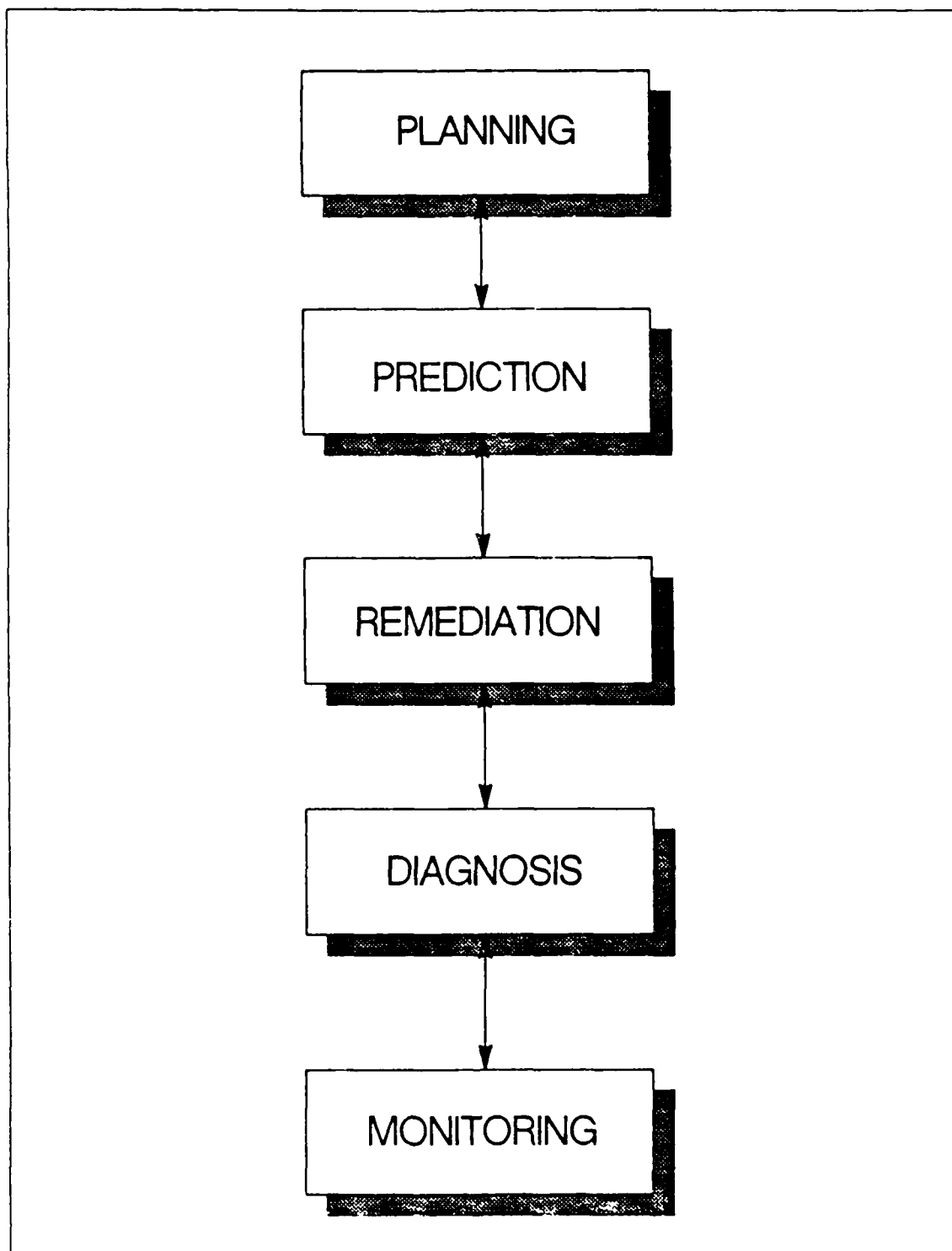


Figure 3. Diagnosis Hierarchy.

To provide a meaningful input to the levels in the diagnostic hierarchy which use symbolic processing, the monitor must also convert its quantitative assessment of the fault situation to a qualitative fault symptom. For example, an engine temperature sensor reading that is 75 degrees higher than the normal range would be reported as "Engine Temperature Too High." This qualitative fault symptom will serve as an input to the diagnosis level of the hierarchy, where the implications of the symptom will be determined.

3.1.2.2 Diagnosis As was discussed in the previous chapter, there are a variety of ways that diagnosis can be accomplished, but they all have the same goal. Given a set of fault symptoms, the diagnosis function must try to determine the root cause of those symptoms.

Ideally, the diagnosis function should isolate a single faulty primitive component which is responsible for all the observed fault symptoms. (In this context, a primitive component is defined as a component that is not made up of other components, and therefore is at the bottom of the functional hierarchy. Primitive components are assembled to form composite components, which themselves can be assembled to eventually form the entire aircraft.) If this is not possible, the next best situation is to isolate the fault to a single composite component. The diagnostic function should move up the functional hierarchy of the aircraft until it finds a level at which it can identify a faulty component responsible for the observed symptoms. By starting at the bottom of the functional hierarchy, the diagnosis function strives to identify the most primitive, and therefore the most specific, component to explain the cause of the observed fault symptoms. Only after it is found that a fault in one of the primitive components cannot account for all observed fault symptoms will the diagnosis function move up one level of the functional hierarchy and attempt to identify a faulty composite component.

A tradeoff occurs when the diagnosis function must move up the functional hierarchy to find a suitable explanation for the fault symptoms. The diagnosis becomes less specific and therefore less useful to the next higher levels in the diagnostic hierarchy. On the other hand, moving up

in the functional hierarchy and becoming less specific tends to increase the probability that a responsible component will be found. Making a less specific diagnosis is better than no diagnosis at all. The tradeoff is beneficial because identifying the malfunctioning component is not the only task performed by the diagnosis function. The diagnosis function also determines the other components in the functional hierarchy whose performance is probably or potentially affected by the faulty component. This ability to not only determine the cause of a set of fault symptoms, but to determine the side effects of the fault, is of great benefit to the flight crew in assessing the overall aircraft status, and is the basis for the next higher levels of the diagnostic hierarchy.

3.1.2.3 Remediation The next logical step after the monitoring function identifies fault symptoms and the diagnosis function determines the underlying fault and its side effects is to recommend the best course of action given the current situation. This is the purpose of the remediation function.

While it may appear simple to "remedy the situation," a remedy may take a number of different forms depending on when it is applied and the intended outcome. Two opposite approaches are to a) compensate for the current set of fault symptoms (treating the symptoms), or b) remove the source of the current set of fault symptoms (treating the causes). Either one of these approaches can be employed for a variety of reasons, including to:

1. Conserve resources,
2. Prevent further malfunctions,
3. Ensure mission accomplishment,
4. Ensure crew safety, or
5. Ensure aircraft safety.

For the purposes of this study, a single remediation approach and a single reason were chosen to be implemented in the NASP system status monitor. It was decided the remediation func-

tion should seek to compensate for the current set of fault symptoms in order to ensure mission accomplishment.

In contrast to the diagnosis function, where fault hypothesis generation used a bottom-up approach on the functional hierarchy, the remediation function should use a top-down approach. In a fault situation, remediation will attempt to deal first with the symptom that is having the most immediate effect on the highest affected level of the functional hierarchy. Since the stated goal of the remediation function is to ensure mission accomplishment, this method will work to relieve the symptom that is most threatening to the mission. From this point, the remediation function should search for the lowest-level, or most primitive, action that will produce the desired effect on the most threatening symptom.

3.1.2.4 Prediction Before the corrective action proposed by the remediation function can be put into effect, the status monitor needs to determine the possible consequences of the proposed action. Although the remedial action is intended to compensate for the detrimental effects of the fault symptoms, it may have other side effects that will make the fault situation worse or produce a completely different fault situation. The new system status resulting from the remedial action must be compared to a status which is normal for the current flight phase. If a fault situation is found in the predicted status, the proposed remedial actions must be discarded. This is the purpose of the prediction function.

It should be stressed that the prediction function will deal only with the immediate consequences of the proposed remedial action. If the prediction function finds the proposed action to be unacceptable, it will request that the remediation function develop a different remedial action for the prediction function to test. This process will continue until an acceptable remedial action is found. At this point, the acceptable action is sent to the planning function.

3.1.2.5 *Planning* The purpose of the planning function is to determine the long-range consequences of the proposed remedial action. The ultimate question to be answered by this function is if the consequences of the remedial action will allow completion of the mission. If the action proposed by the remediation function is consistent with the mission objectives, that action will be presented to the flight crew for their approval. If the proposed action jeopardizes any aspect of the mission, the action will be rejected and the remediation function will be asked to propose a different action. If the proposed action is acceptable, it will be carried out. Depending on the circumstances, an acceptable action may produce a wide range of outcomes. On one hand, the action may allow the mission to be completed with all objectives met. At the other extreme, the best course of action may be to abort the mission and "cut the losses." The planning function should pick the best alternative while working within any constraints imposed by considerations such as safety, cost, security, etc.

Figure 4 shows the sequence of steps that the system status monitor takes in trying to resolve an observed fault situation.

3.2 *Semantic Network Knowledge Representation*

The physical and functional relationships that make up the National Aerospace Plane domain are organized in a semantic network representation. This representation is virtually the same as is used in the Faultfinder system developed at NASA Langley Research Center. (Faultfinder will be discussed further in the Chapter IV.) However, a number of changes and additions were made to accommodate the additional capabilities of the NASP system status monitor.

Semantic networks were originally developed as a way of representing the meaning of English words [18, 215]. The objects to be represented are the nodes of the network, and the relationships between the objects are the arcs connecting the nodes. Each arc has a direction to signify the direction of the relationship. Two-way relationships must be expressed explicitly

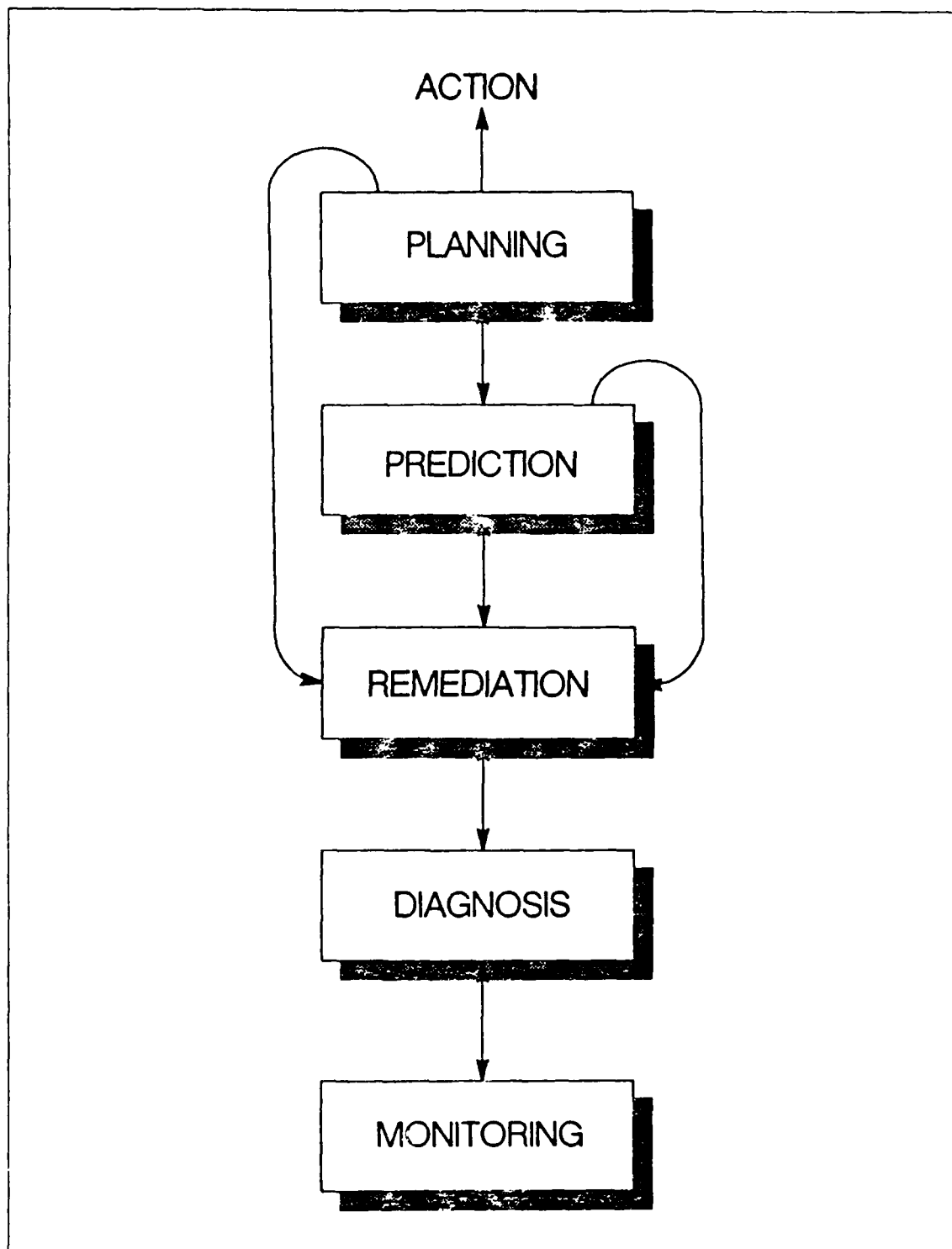


Figure 4. Steps in the Diagnostic Process.

Specific objects in a semantic network can be shown to belong to a general class of objects through an IS-A relationship. That is to say that the object "is a" specific instance of the general class of objects. Figure 5 shows the object classes used in the NASP system status monitor knowledge base and the specific object instances in each of those classes of objects.

The content of a particular semantic network not only depends on the objects to be represented, but also on the reasoning to be applied to the network. As an example, reasoning about the parts that make up a device would require arcs named PARTS from the device object to the individual part objects. The diagnosis function of the NASP system status monitor incorporates reasoning about the physical make-up and functional dependencies of the NASP aircraft. Therefore, the knowledge base in the system status monitor is represented in those terms. Figure 6 shows the relationships used in the NASP system status monitor knowledge base.

The remediation function of the NASP system status monitor performs reasoning about actions that will produce changes in the observed values of sensors. Therefore, the NASP knowledge base also includes causal information to facilitate this reasoning. The next section describes this causal reasoning representation.

3.3 Causal Knowledge and Reasoning

The knowledge used by the remediation function of the NASP system status monitor is contained in the semantic network knowledge base and is associated with the sensor objects. The intent is to represent a set of actions that will cause a predictable change in the sensor reading. This usually involves altering the conditions that the sensor is measuring. As an example, the airspeed sensor measures airspeed. The causal knowledge attached to the airspeed sensor in the knowledge base will include those actions that can affect airspeed. These would include increasing or decreasing thrust, increasing or decreasing drag, etc.

Causal reasoning in this system involves chaining together a series of cause and effect pairs.

<u>Object Class</u>	<u>Object Instances</u>
Mission	Mission
Flight-Phase	Takeoff Climb Cruise Descent Landing
Flight-Parameter	Total-Thrust Weight Drag Attitude Lift
Aircraft-Sensor	AirspeedA AltitudeA Climb-RateA MachA Sink-RateA PitchA RollA YawA
Plane	Takeoff-Plane Climb-Plane Cruise-Plane Descent-Plane Landing-Plane
Aircraft-System	Propulsion-SystemA Hydraulic-SystemA Fuel-SystemA Flight-Control-SysA Thermal-Protection-SysA
Engine	EngineA EngineB
Engine-Sensor	N1A, B EprA, B EgtA, B VoltageA, B ThrustA, B VibrationA, B

Figure 5. Object Classes in the Semantic Network Knowledge Base and the Object Instances in Each Class.

<u>Object Class</u>	<u>Object Instances</u>
Engine Component	InletA, B CompressorA, B GearboxA, B Electric-GeneratorA, B Gas-GeneratorA, B TurbineA, B Fuel-InjectorA, B CombustorA, B NozzleA, B
Hydraulic-Subsystem	Hydraulic-SubsystemA Hydraulic-SubsystemB
Hydraulic-Line	Hydraulic-LineA Hydraulic-LineB
Hydraulic-Pump	Engine-Hyd-PumpA1 Engine-Hyd-PumpB1 Electric-Hyd-PumpA2 Electric-Hyd-PumpB2
Hyd-Pressure	Hyd-PumpA1-Pressure Hyd-PumpA2-Pressure Hyd-PumpB1-Pressure Hyd-PumpB2-Pressure
Hydraulic-Reservoir	Hydraulic-ReservoirA Hydraulic-ReservoirB
Hyd-Quantity	Hyd-QuantityA Hyd-QuantityB
Fuel-Tank	Fwd-Fuel-Tank Aft-Fuel-Tank EngineA-Feed-Tank EngineB-Feed-Tank
Fuel-Pump	Fwd-Tank-Transfer-Pump Aft-Tank-Transfer-Pump EngineA-Tank-Boost-Pump EngineB-Tank-Boost-Pump

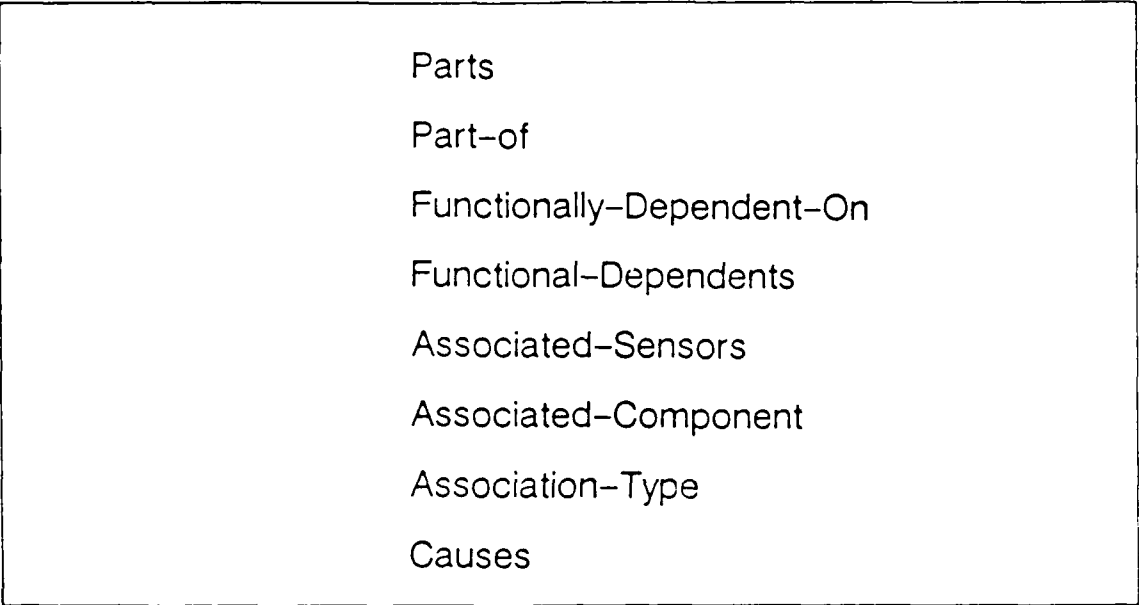
Figure 5. Object Classes in the Semantic Network Knowledge Base and the Object Instances in Each Class (continued).

<u>Object Class</u>	<u>Object Instances</u>
Fuel-Valve	Crossfeed-Valve Fuel-Dump-ValveA Fuel-Dump-ValveB
Fuel-Line	Fuel-LineA Fuel-LineB
Fuel-Flow	Fuel-FlowA Fuel-FlowB
Fuel-Qty-Sensor	Fwd-Tank-Quantity Aft-Tank-Quantity Feed-TankA-Quantity Feed-TankB-Quantity Total-Fuel-Quantity Fuel-Imbalance
Control-Surface	Left-Elevon Right-Elevon Body-Flap Rudder
Control-Surface-Actuator	Left-Elevon-Actuator-1, 2 Right-Elevon-Actuator-1, 2 Body-Flap-Actuator-1, 2 Rudder-Actuator-1, 2
Control-Surface-Position	Left-Elevon-Position Right-Elevon-Position Body-Flap-Position Rudder-Position
Cooling-Subsystem	Nosecap-Cooling Left-Wing-Cooling Right-Wing-Cooling Engine-Inlet-Cooling EngineA-Internal-Cooling EngineB-Internal-Cooling Engine-Nozzle-Cooling Vert-Tail-Cooling

Figure 5 Object Classes in the Semantic Network Knowledge Base and the Object Instances in Each Class (continued).

<u>Object Class</u>	<u>Object Instances</u>
Cooling-Pump	Fwd-Hyd-Cooling-Pump Fwd-Elec-Cooling-Pump Left-Hyd-Cooling-Pump Left-Elec-Cooling-Pump Right-Hyd-Cooling-Pump Right-Elec-Cooling-Pump
Temp-Sensor	Nosecap-Temp Left-Wing-Temp Right-Wing-Temp Engine-Inlet-Temp EngineA-Internal-Temp EngineB-Internal-Temp Engine-Nozzle-Temp Vert-Tail-Temp
Cooling-Pressure	Fwd-Hyd-Cooling-Pressure Fwd-Elec-Cooling-Pressure Left-Hyd-Cooling-Pressure Left-Elec-Cooling-Pressure Right-Hyd-Cooling-Pressure Right-Elec-Cooling-Pressure

Figure 5. Object Classes in the Semantic Network Knowledge Base and the Object Instances in Each Class (continued).



Parts
Part-of
Functionally-Dependent-On
Functional-Dependents
Associated-Sensors
Associated-Component
Association-Type
Causes

Figure 6. Semantic Network Relationships.

The goal is to reach, at the end of the chain, the most fundamental action that will ultimately cause the desired change in the sensor reading at the head of the chain.

The causal reasoning process can best be explained with an example. If the NASP mission is being threatened by a low climb rate in the climb flight phase, something must be found to increase the climb rate. One option is to increase engine thrust. So now a further action must be found to increase thrust. This chaining process will continue until finding the most elementary action which will produce the desired result.

Since several different chains of actions may produce the same desired result, some method must be employed to decide which actions to choose. Some logical alternatives are to choose:

1. Actions that most directly affect the diagnosed fault component.
2. Actions which counteract the greatest number of fault symptoms.
3. Actions which themselves expend the least resources, etc.

For the NASP system status monitor, alternative 2 was selected for choosing the appropriate remedial action.

IV. Program Development

The theoretical basis for the National Aerospace Plane (NASP) system status monitor was explained in the previous chapter. This chapter will describe the development of a computer program prototype for a NASP system status monitor that implements those theories.

The first section of this chapter lists the alternatives explored for a suitable development environment for the prototype system status monitor. The next section describes the knowledge base used to represent the NASP domain. The last three sections outline the monitoring, diagnosis, and remediation functions which use the knowledge base.

4.1 Possible Programming Approaches

Several computer programming techniques were explored for the development of the prototype NASP system status monitor. These included off-the-shelf expert system shells, blackboard system shells, and a dedicated aircraft diagnosis system called Faultfinder which was ultimately chosen.

4.1.1 Expert System Shells The first approach investigated for implementation of the system status monitor was standard expert system shells. The two systems most seriously considered were the Automated Reasoning Tool (ART) developed by Inference Corporation, and the Knowledge Engineering Environment (KEE) developed by Intellicorp. Both of these systems offer a very rich development environment, with excellent editing and debugging facilities. ART is primarily rule-based, while KEE uses a frame- and object-oriented knowledge representation. Either one of these systems could have been an adequate method with which to implement a system status monitor, but neither directly supported the blackboard problem-solving model which was an original requirement of this project. Therefore, neither ART nor KEE was considered the first choice for the system status monitor development tool.

4.1.2 Blackboard Shells Three blackboard shells were considered as NASP system status monitor programming tools. These were BB1 (Blackboard One) developed at Stanford University [7], ABE (A Better Environment) developed by Teknowledge Corporation [6,11], and SCHEMER developed by Dr. Michael Fehling of Rockwell International Science Center.

All three blackboard shells offered the ability to integrate the reasoning of separate knowledge sources. This capability corresponded well with the diagnostic hierarchy model. Each level of the hierarchy could have been implemented as a separate knowledge source, and the functional hierarchy could have been mapped into a multi-level blackboard. However, none of the blackboard shells was available. BB1 was ordered from Stanford University in May 1987, but its delivery date was uncertain and so it could not be considered the primary implementation choice. Both ABE and SCHEMER were still in development during the summer of 1987, making them unavailable for this project.

4.1.3 Faultfinder Research by Kathy Abbott and Paul Schutte in the Vehicle Operations Research Branch, NASA Langley Research Center, was directed toward real-time fault monitoring and diagnosis for commercial transport aircraft. Their work had the following objectives relative to aircraft onboard fault monitoring and diagnosis:

1. Identify guidelines for automation,
2. Identify crew interfaces,
3. Determine if artificial intelligence techniques could be used, and,
4. Develop a prototype to demonstrate the chosen approach. [19, 1]

The prototype system they developed is called Faultfinder. It includes fault monitoring and diagnosis functions, and a blackboard structure to pass information between the functions. The fault monitor is based on a numerical model of the JT8D turbojet engine. The monitor can either input data from a stored time-ordered file of sensor readings, or it can interactively accept fault

symptoms from the operator. Fault symptoms either computed or accepted by the monitor are passed to the diagnosis system, which performs its function in two stages.

Stage 1 of the diagnosis function performs fault-symptom association. This is a rudimentary rule-based approach which matches the current fault symptoms with the condition part, or left-hand side, of a set of rules. If a match is found, the action part, or right-hand side, of the matched rule is reported as the cause of the fault symptoms. Stage 1 has no chaining capability, and so cannot use the rules to produce intermediate conclusions. If a match is not found by Stage 1, the fault symptoms are passed to Stage 2.

Stage 2 uses the fault symptoms and model-based reasoning to localize the fault and produce a fault hypothesis. The model is a semantic network representation of the aircraft's functional and physical structure. To produce a valid fault hypothesis, Stage 2 generates many interim hypotheses, each of which begins with the assumption that one of the aircraft's primitive components is responsible for all the current fault symptoms. Each hypothesis is produced by building a chain of dependency from the primitive component through all those components that depend on it. This dependency chain is called a propagation path in Faultfinder. The propagation path stops if a) a component is reached that has no other components depending on it (usually the top of the component hierarchy), or b) a component is reached which has a sensor associated with it and the sensor is not one that is producing one of the current fault symptoms. A hypothesis produced in this way is considered to be valid if all the current symptoms come from sensors that are associated with one of the components on this hypothesis' propagation path.

Components on the propagation path of a valid hypothesis are assigned different degrees of fault severity. The primitive component at the beginning of the propagation path is called the RESPONSIBLE-COMPONENT. Components whose associated sensors are producing the current symptoms are called DEFINITELY-AFFECTED. Components which are on the propagation path but do not have associated sensors are called POSSIBLY-AFFECTED. These three degrees of

fault severity are presented to the operator in a series of graphics displays which show drawings of the overall aircraft and its individual systems. On the graphics displays, the drawing of the RESPONSIBLE-COMPONENT is shaded darkest, and the POSSIBLY-AFFECTED components are shaded lightest. These displays quickly show the operator the direct and indirect effects of the current fault situation.

The Faultfinder system closely matches the requirements for the NASP system status monitor. It already has two of the five diagnostic hierarchy levels, their interfaces are implemented through a blackboard, and the semantic network knowledge base has the structure needed to develop the full functional hierarchy. For these reasons, Faultfinder was chosen to serve as the basis for the NASP System Status Monitor.

4.2 NASP Knowledge Base

The first task in modifying the Faultfinder system to become a NASP System Status Monitor (SSM) was to develop the knowledge base. This involved both making the knowledge base specific to the NASP domain and extending the knowledge base to include all five levels of the functional hierarchy.

4.2.1 NASP-specific Knowledge Faultfinder's knowledge base originally contained representations of only the hydraulic system and one engine. There were functional and physical dependency links within those two systems, but neither of those links existed between the system and aircraft levels. The only links that existed between these levels showed that one was a PART-OF the other.

The NASP aircraft description first needed different system definitions than those used in Faultfinder. The scope of this study did not allow an exhaustive description of every possible system in an aircraft as complex as the NASP. Therefore, a subset of five primary systems was chosen to represent the NASP aircraft. These five systems are:

1. Propulsion system, which includes two engines.
2. Hydraulic system with two independent subsystems.
3. Fuel system,
4. Flight control system, and
5. Thermal protection system.

Appendix A contains a description of the function and structure of each of these systems.

After the five aircraft systems and their components were added to the knowledge base, their interconnections were represented with functional dependency links. Here, the difference between Faultfinder and the NASP SSM is that the NASP knowledge base shows that the aircraft is dependent on the proper functioning of its constituent systems, while Faultfinder does not. In the NASP SSM, inter-level dependency extends from the top of the functional hierarchy to the bottom. It is this dependency between levels that allows the NASP SSM to show how faults at any level of the hierarchy can affect any higher level.

4.2.2 Extending the Knowledge Base The last additions to the NASP SSM knowledge base were the two highest levels of the hierarchy: the mission and flight phase levels. The structure of these two levels is somewhat different than the lower three levels. These two levels have parts and functional dependencies that are conceptual operating states rather than physical hardware. As an example, the mission itself is an extended operating state, and it is dependent on the five flight phases, which are also operating states. In turn, each of the flight phases is functionally dependent on both physical (the aircraft) and conceptual (lift, drag, altitude, etc.) components.

One shortcoming of the structure of the knowledge base is its inability to represent logical relationships. If a component is dependent on three other components, there is no way to say that it depends on all three at the same time (1 AND 2 AND 3), or that in some situations it depends on two components together or a third by itself ((1 AND 2) OR 3).

The complete NASP System Status Monitor Knowledge Base is listed in Appendix B.

4.3 Monitoring Function

Faultfinder offers two different ways to perform the monitoring function. First, the user can provide a file of raw sensor readings. These data will be read by Faultfinder, and the values will be compared to the sensor values predicted by the monitor's numerical models. If the input data disagree with the predicted sensor values, the monitor will produce fault symptoms to be used by the diagnosis function. In the alternate method, the user can interactively enter fault symptoms, thus bypassing the numerical models.

The NASP SSM currently allows only interactive entry of fault symptoms. The user selects the system where a symptom is to appear, the sensor which will report the symptom, and the qualitative value reported by the sensor. Figure 7 shows the available systems in the NASP SSM, the sensors that each system contains, and the values that can be assigned to each sensor.

As Figure 7 indicates, the user is not able to specify the time-variance of any of the sensor values, only the current value. The addition of this capability would allow the SSM to perform temporal reasoning tasks, such as prediction and planning.

When the NASP and its missions are more clearly defined, numerical models of its systems and operations can be developed and added to the NASP SSM monitoring function.

4.4 Diagnosis Function

Both Faultfinder and the NASP SSM perform a two-stage diagnosis function and display their results in both text and graphics form. However, there is one important difference in the way the NASP SSM performs its second stage. This difference allows the NASP SSM to determine the propagation of fault affects through the entire functional hierarchy, not just within a single system.

<u>System</u>	<u>Sensor</u>	<u>Values</u>
Aircraft	AirspeedA	High, Normal, Low
	AltitudeA	
	Climb-RateA	
	MachA	
	Sink-RateA	
	PitchA	
	RollA	
	YawA	
Engine (left or right)	N1	High, Normal, Low
	Epr	
	Egt	
	Voltage	Yes, No
	Thrust	
	Vibration	
Hydraulic-Subsystem (A or B)	Hyd-Subsys-Pressure	High, Normal, Low
	Hyd-Pumpl-Pressure	
	Hyd-Pump2-Pressure	
Fuel-SystemA	Fuel-FlowA	High, Normal, Low
	Fuel-FlowB	
	Fwd-Tank-Quantity	
	Aft-Tank-Quantity	
	Feed-TankA-Quantity	
	Feed-TankB-Quantity	
	Total-Fuel-Quantity	
	Fuel-Imbalance	
Flight-Control-System	Left-Elevon-Position	High, Normal, Low
	Right-Elevon-Position	
	Body-Flap-Position	
	Rudder-Position	
Thermal-Protection-Sys	Nosecap-Temp	High, Normal, Low
	Left-Wing-Temp	
	Right-Wing-Temp	
	Engine-Inlet-Temp	
	EngineA-Internal-Temp	
	EngineB-Internal-Temp	
	Engine-Nozzle-Temp	
	Vert-Tail-Temp	
	Fwd-Hyd-Cooling-Pressure	
	Fwd-Elec-Cooling-Pressure	
	Left-Hyd-Cooling-Pressure	
	Left-Elec-Cooling-Pressure	
	Right-Hyd-Cooling-Pressure	
	Right-Elec-Cooling-Pressure	
	Right-Elec-Cooling-Pressure	

Figure 7. Systems, Sensors, and Sensor Values.

4.4.1 Stage 1 Both Faultfinder and the NASP SSM use a two-stage fault diagnosis process. Stage 1 compares the current symptoms to a set of stored fault-symptom association rules. This diagnosis stage has the advantage of quickly recognizing the most common fault situations. If the fault symptoms do not match any of the rules in Stage 1, then Stage 2 of the diagnosis function is engaged.

4.4.2 Stage 2 To diagnose a fault situation, Faultfinder's Stage 2 produces a series of diagnosis hypotheses. Each hypothesis consists of a list of components from the functional hierarchy of the aircraft. This list is called the propagation path, and starts with a unique primitive component. Each hypothesis is based on the assumption that its primitive component is responsible for the current set of fault symptoms. This assumption is tested by building a propagation path from the primitive component to each component that is dependent on it (as determined by the "functional-dependents" links in the semantic network). A propagation path is stopped in one of two ways. The first and most obvious way is if the propagation path reaches the top of the functional hierarchy. The second way is more subtle and also more important. If the propagation path reaches a component that has a sensor associated with it (as determined by the "associated-sensor" link in the semantic network), and that sensor is not producing one of the current fault symptoms, then the propagation path stops. This reason for stopping a propagation path is the basis of the diagnosis process and deserves further explanation.

By assuming that a particular primitive component is responsible for the current fault situation, the diagnosis function also assumes that the effects of the faulty primitive component will propagate through the functional hierarchy. Since the diagnosis function only handles single faults, then all current fault symptoms must be caused by the propagated effects of the responsible component. For a fault to propagate, its effects must be felt on the entire propagation path. Therefore, if a component has a sensor which is not affected by the fault propagation, that component cannot be on the propagation path.

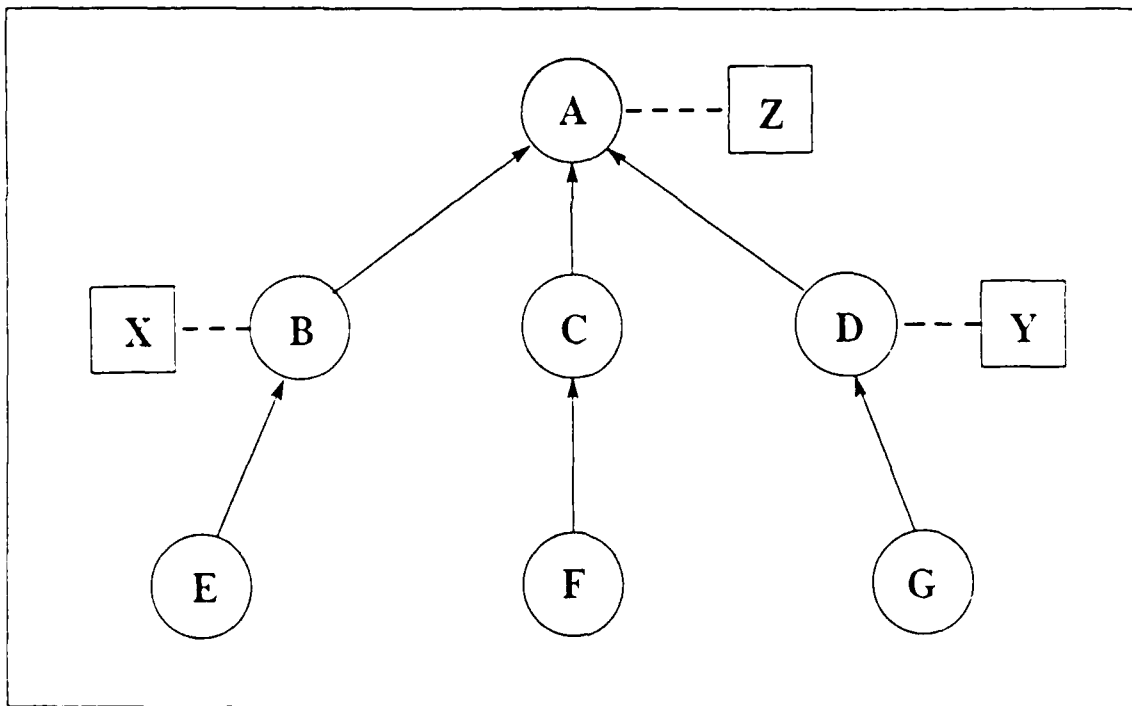


Figure 8. Sample Fault Propagation Tree.

Once hypotheses are produced for all the primitive components, they are tested for validity. A hypothesis is valid only if it explains all of the current fault symptoms. That is, the propagation path of a valid hypothesis will contain all the components whose sensors are producing the current fault symptoms.

A simple example will help to illustrate this process. Consider the graph in Figure 8. Nodes "A" through "G" represent the components of a system being diagnosed, with nodes "E", "F" and "G" representing primitive components. The nodes are connected by directed arcs which represent functional dependencies. Thus, "B" depends on "E", "C" depends on "F", "D" depends on "G", and "A" depends on "B", "C" and "D". There are three sensors in this system: "X", "Y" and "Z", with sensors "Y" and "Z" reporting faults.

The diagnosis function will attempt to build a path from a primitive component through all the components whose sensors are affected. Each attempt will be a hypothesis. The first hypothesis

will begin with a path from "E" to "B". Since the sensor associated with "B" is not affected, this path cannot be completed, and this hypothesis will include only "E".

The second hypothesis will begin with a path from "F" to "C". Since there is nothing at "C" to stop the propagation, this path will continue to "A", which has an affected sensor and so should be included. Therefore, the second hypothesis is "F-C-A(X)".

The third hypothesis will begin with a path from "G" to "D". Node "D" has an affected sensor, so it can be included in the propagation path. The next node, node "A", also has an affected sensor and can be included in the path. Therefore, the third hypothesis is "G-D(Y)-A(X)".

After all possible hypotheses have been produced, the diagnosis function will determine if any of the hypotheses is valid. The test for validity will be if the hypothesis includes all of the currently affected sensors. Of the three hypotheses produced in this example, only the third hypothesis includes both sensors "X" and "Y". Therefore, the third hypothesis is the only valid hypothesis, and it has declared the primitive component at node "G" to be responsible for the current fault.

There is one major difference between Faultfinder's implementation of the diagnosis function and the NASP SSM's implementation. When Faultfinder is activated, it reads a file containing the physical description of the aircraft. This description is in the form of the semantic network knowledge base. In the knowledge base, the sensors are linked to the components to which they are physically attached. Faultfinder modifies the knowledge base after it is loaded so that the sensor associations are "migrated" up the functional hierarchy. This has the affect of giving any particular component a list of associated sensors that includes its own original sensors and all sensors from its constituent parts. This sensor migration arrangement has some practical uses, such as localizing the generation of hypotheses to only those parts of the knowledge base that have affected sensors. However, sensor migration has a detrimental effect on the form of diagnosis used in the NASP SSM.

Faultfinder's knowledge base has functional dependency links only within systems. Therefore, fault effects can propagate only within a system. On the other hand, the NASP SSM's functional

dependency links extend from the bottom to the top of the functional hierarchy in order to show the full effect of a fault situation. If the NASP SSM were to use the sensor migration technique, the component at the top of the functional hierarchy would have every sensor in the physical system associated with it. Therefore, the top component could be on every propagation path, and far too many seemingly valid hypotheses would be produced. For this reason, the NASP SSM does not use sensor migration, and so does not localize its hypothesis generation. What the NASP SSM loses in efficiency is gained in its ability to show the full effects of a fault situation.

Both Faultfinder and the NASP SSM produce a default hypothesis if a valid hypothesis cannot be generated. A default hypothesis will consist of two or more separate, unconnected fault propagation paths. Each of these default paths will begin with a component with an affected sensor. This component is not necessarily a primitive component. The default paths will propagate from these components until stopping for one of the reasons stated above.

If all the primitive components failed to produce a valid hypothesis, an alternative to producing a default hypothesis would be to attempt to build hypotheses based on composite components. This would allow Stage 2 to narrow the diagnosis to a subsystem or system rather than a primitive component. This capability should be explored as an enhancement to the NASP SSM.

Whether it produces valid or default hypotheses, the diagnosis function displays its results both as text and graphics. The diagnosis graphics displays will be examined next.

4.4.3 Diagnosis Displays Figure 9 shows the NASP SSM system display. The system display is divided into four windows, or panes, where different information is presented. Monitor information is displayed in the upper right and lower left panes. The upper right pane shows a graphical representation of engine instruments. The instrument readings change with changes in input sensor data. A future enhancement to the NASP SSM would have instrument displays for the other aircraft systems displayed in this pane at appropriate times. The current fault symptoms are listed in the lower left pane.

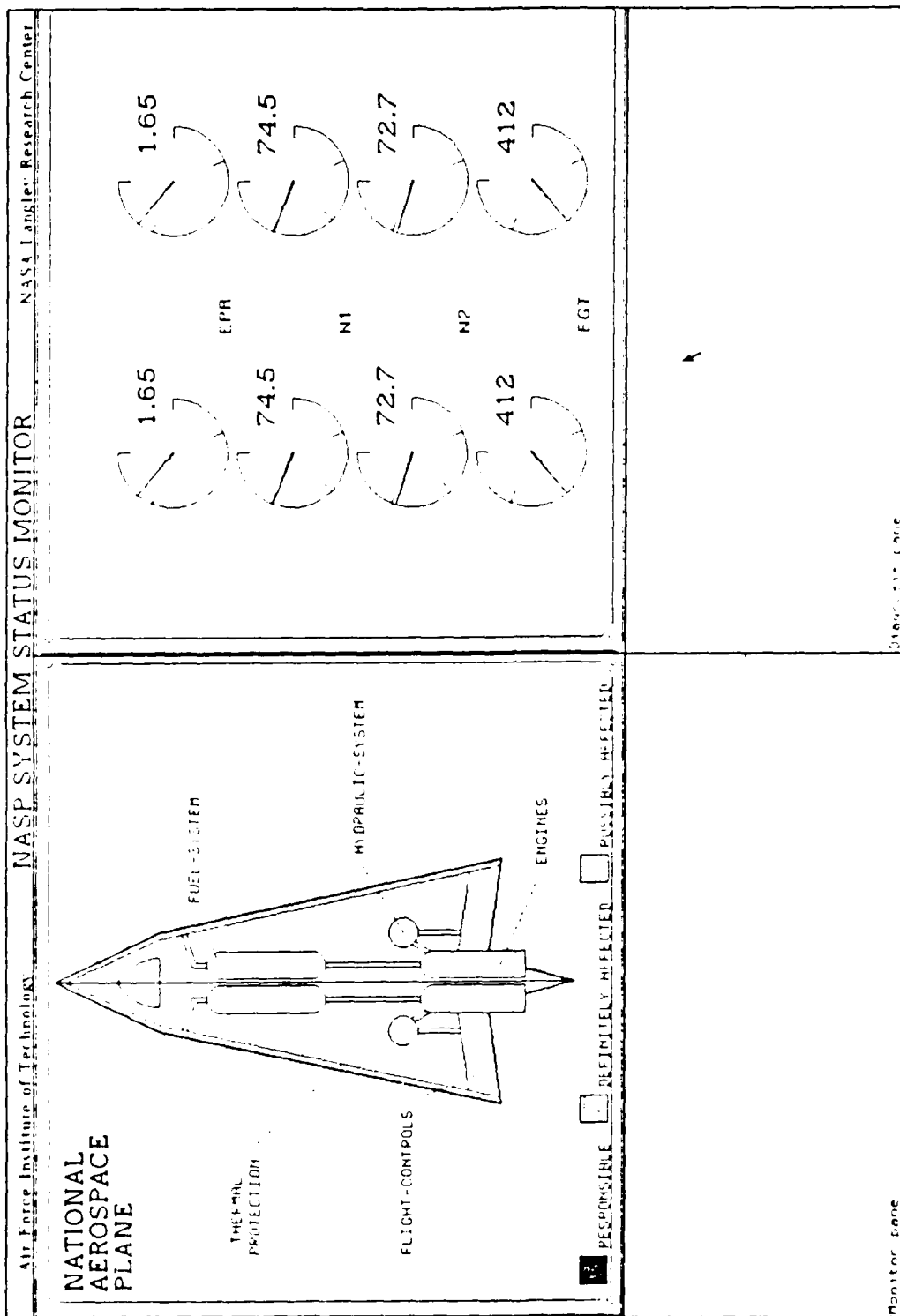


Figure 9 NASP SSM System Display

Results of the diagnosis function occupy the upper left and lower right display panes. After the diagnosis function has produced a valid or default hypothesis, its results are displayed in two ways. First, the hypothesis is listed in textual form in the lower right display pane, called the diagnosis pane. The diagnosis pane is too small to display the entirety of most hypotheses, so the display is scrolled one pane at a time. Figure 10 shows an example of the entire listing of a fault hypothesis. The first line of the hypothesis listing shows which hypothesis is being listed if there are more than one. The next section, labeled "Causes," shows the results of Stage 1 of the diagnosis function. If no Stage 1 Diagnosis has been produced, the cause will be listed as "Unknown." Next, each affected component in the fault propagation path is listed along with the component's fault severity. Fault severities fall into three categories. A "RESPONSIBLE-COMPONENT" is the component judged to be responsible for all the current fault symptoms. A "DEFINITELY-AFFECTED" component is one that is directly on the fault propagation path, or one that has an affected sensor. A "POSSIBLY-AFFECTED" component is one that is on a branch of the propagation path and has no sensors associated with it. The next section in the hypothesis listing is the type of reasoning used to arrive at the current hypothesis. The possible types are "SINGLE FAULT FUNCTIONAL PROPAGATION" and "SINGLE FAULT PHYSICAL PROPAGATION." The NASP SSM only supports functional propagation. Finally, the fault propagation path is listed. This is the same as the affected components listing, but fault severities are not included.

The results of a fault hypothesis are also displayed graphically in the upper right portion of the display, called the system window. There are 16 different displays that can be shown in the system window. These displays can be grouped into the five levels of the functional hierarchy, as shown in Figure 11. Each display depicts components of its corresponding level of the functional hierarchy. When a fault hypothesis determines that a component is affected by the current fault situation, the outline of that component will be shaded, using the key at the bottom of the system window. The shading corresponds to the fault severity for that component. This shading scheme quickly shows the flight crew those components affected by a fault situation. Figures 12 through 27

Hypothesis 1 of 2

Causes
UNKNOWN

Affected Components

("GEARBOXA" "RESPONSIBLE-COMPONENT")
("GEARBOXA" "DEFINITELYAFFECTED")
("ENGINE-HYD-PUMPA1" "DEFINITELYAFFECTED")
("HYDRAULIC-LINEA" "DEFINITELYAFFECTED")
("HYDRAULIC-SUBSYSTEMA" "DEFINITELYAFFECTED")
("HYDRAULIC-SYSTEMA" "DEFINITELYAFFECTED")
("TAKEOFF-PLANE" "DEFINITELYAFFECTED")
("TAKEOFF" "DEFINITELYAFFECTED")
("MISSIONA" "POSSIBLYAFFECTED")
("CLIMB" "DEFINITELYAFFECTED")
("CLIMB-PLANE" "POSSIBLYAFFECTED")
("CRUISE-PLANE" "POSSIBLYAFFECTED")
("DESCENT-PLANE" "POSSIBLYAFFECTED")
("LANDING-PLANE" "POSSIBLYAFFECTED")
("HYD-SUBSYSA-PRESSURE" "DEFINITELYAFFECTED")
("LEFT-ELEVON-ACTUATOR-1" "DEFINITELYAFFECTED")
("RIGHT-ELEVON-ACTUATOR-1" "POSSIBLYAFFECTED")
("BODY-FLAP-ACTUATOR-1" "POSSIBLYAFFECTED")
("RUDDER-ACTUATOR-1" "POSSIBLYAFFECTED")

Fault Type

Single-Fault-Functional-Propagation

Propagation Path

("GEARBOXA")
("ENGINE-HYD-PUMPA1")
("HYDRAULIC-LINEA")
("HYDRAULIC-SUBSYSTEMA")
("HYDRAULIC-SYSTEMA")
("TAKEOFF-PLANE")
("TAKEOFF")
("MISSIONA")
("CLIMB")
("CLIMB-PLANE")
("CRUISE-PLANE")
("DESCENT-PLANE")
("LANDING-PLANE")
("HYD-SUBSYSA-PRESSURE")
("LEFT-ELEVON-ACTUATOR-1")
("RIGHT-ELEVON-ACTUATOR-1")
("BODY-FLAP-ACTUATOR-1")
("RUDDER-ACTUATOR-1")

Figure 10 Sample Fault Hypothesis Listing

Mission	Mission
Flight Phase	Takeoff
	Climb
	Cruise
	Descent
	Landing
Aircraft	Aircraft
System	Propulsion System
	Hydraulic System
	Fuel System
	Flight Control System
	Thermal Protection System
Subsystem	Left Engine
	Right Engine
	Left Hydraulic Subsystem
	Right Hydraulic Subsystem

Figure 11. Functional Hierarchy Levels and Their Displays.

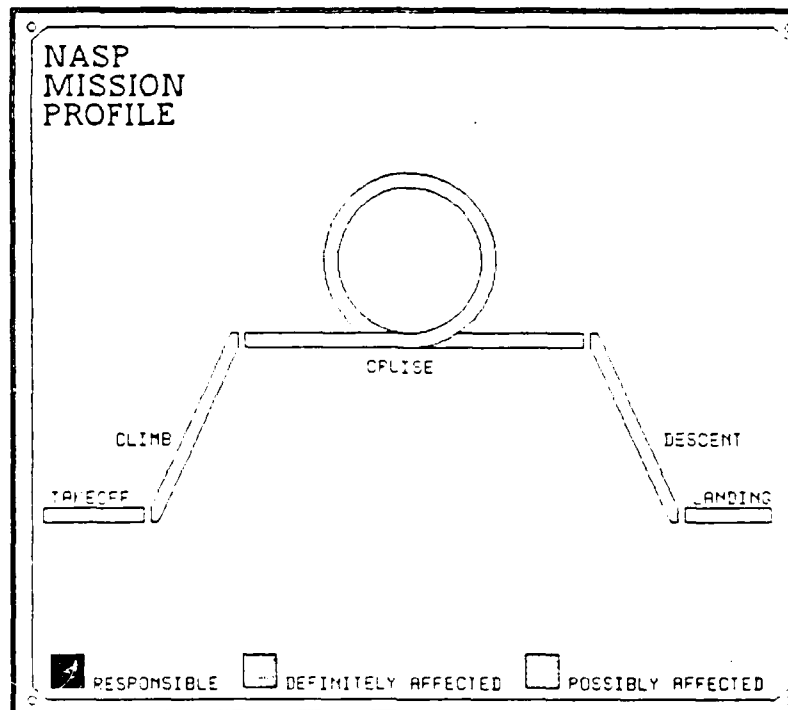


Figure 12. Mission Display.

show each of the 16 possible displays in the system window.

4.5 Remediation Function

The remediation function is intended to propose a course of action to the flight crew that will counteract the effects of the current fault symptoms. As was explained in the previous chapter, it was decided that the remediation function would seek to compensate for the effects of the highest-level fault symptom. The highest-level fault symptom is defined as the symptom whose associated component is highest in the functional hierarchy. The remediation function will attempt to produce one or more remedies for each valid hypothesis.

After the diagnosis function has produced a set of valid hypotheses, the remediation function seeks the highest-level fault symptom. It starts at the top of the functional hierarchy and searches downward until it finds a component whose associated sensor is producing one of the current fault

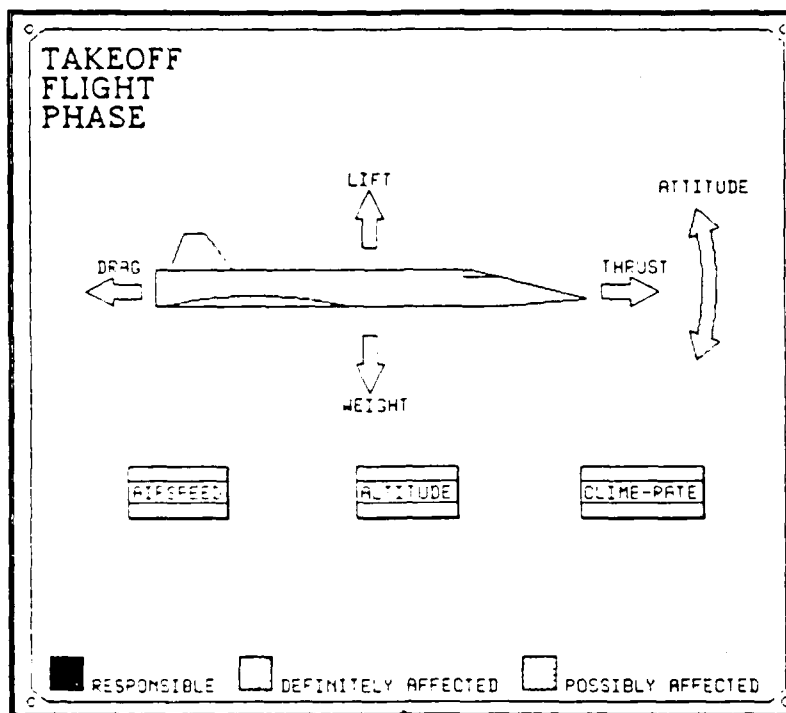


Figure 13. Takeoff Display.

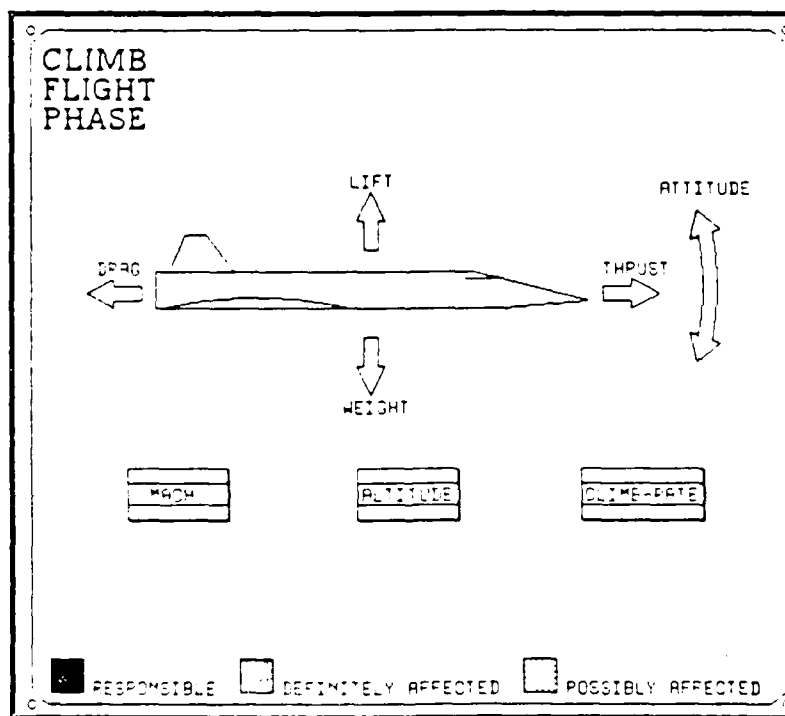


Figure 14. Climb Display.

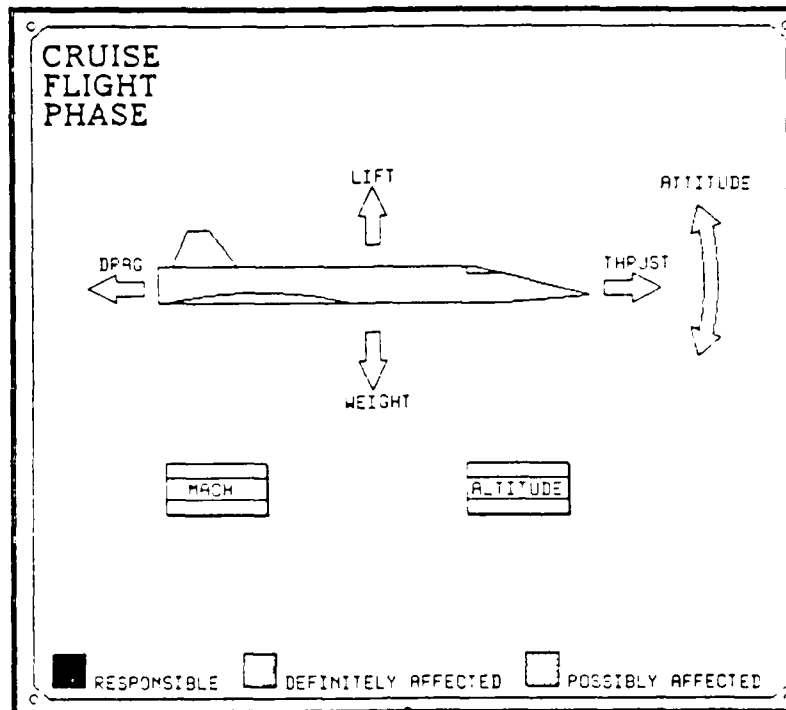


Figure 15. Cruise Display.

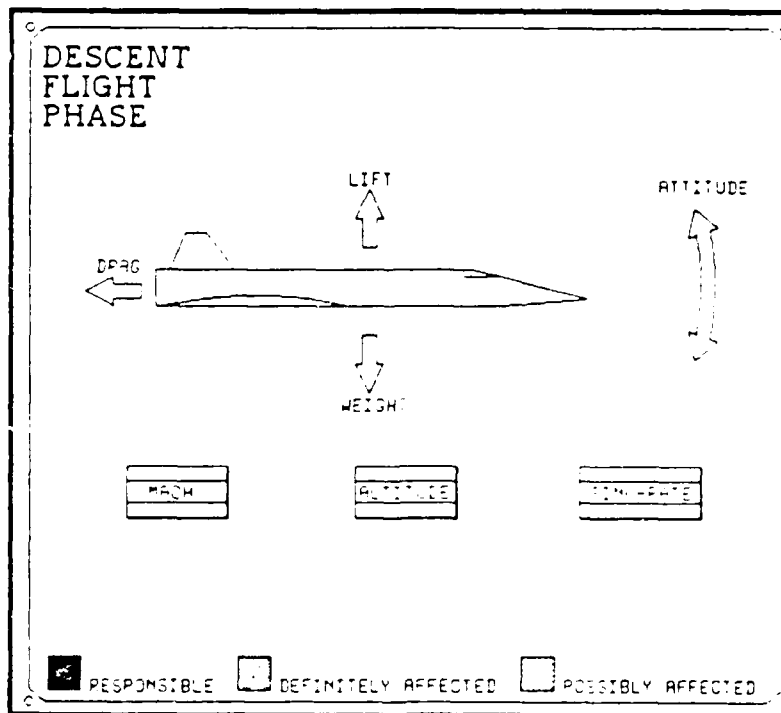


Figure 16. Descent Display.

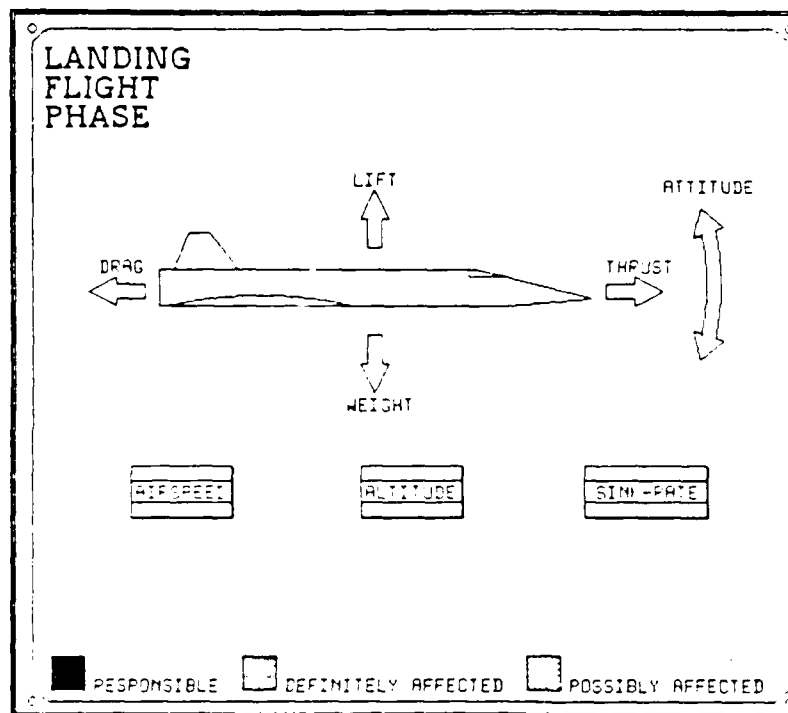


Figure 17. Landing Display.

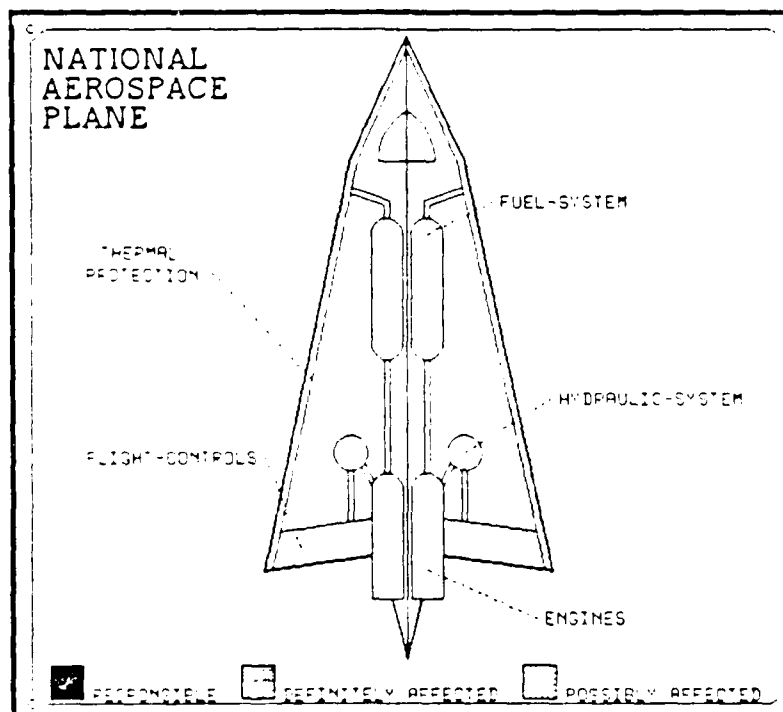


Figure 18. NASP Aircraft Display.

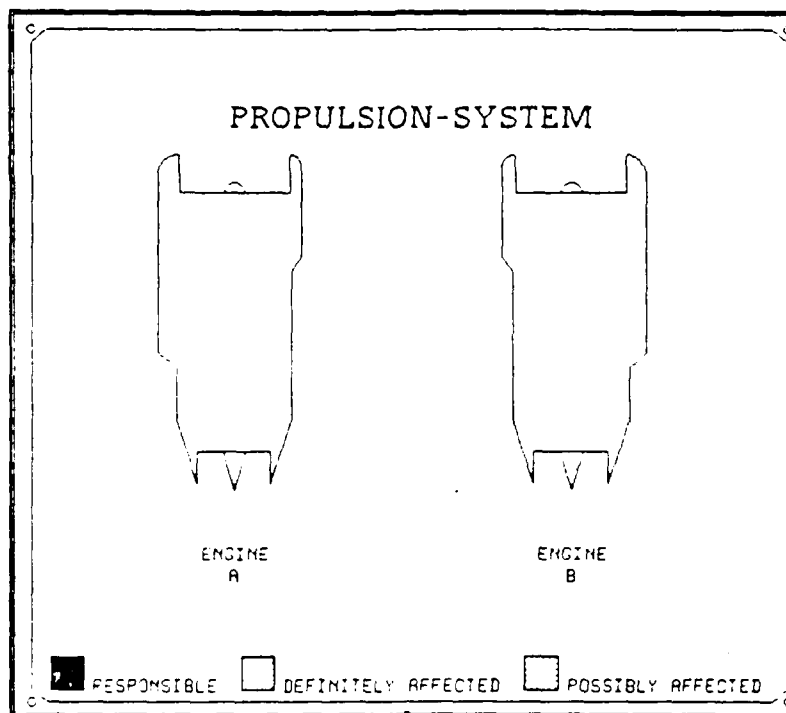


Figure 19. Propulsion System Display.

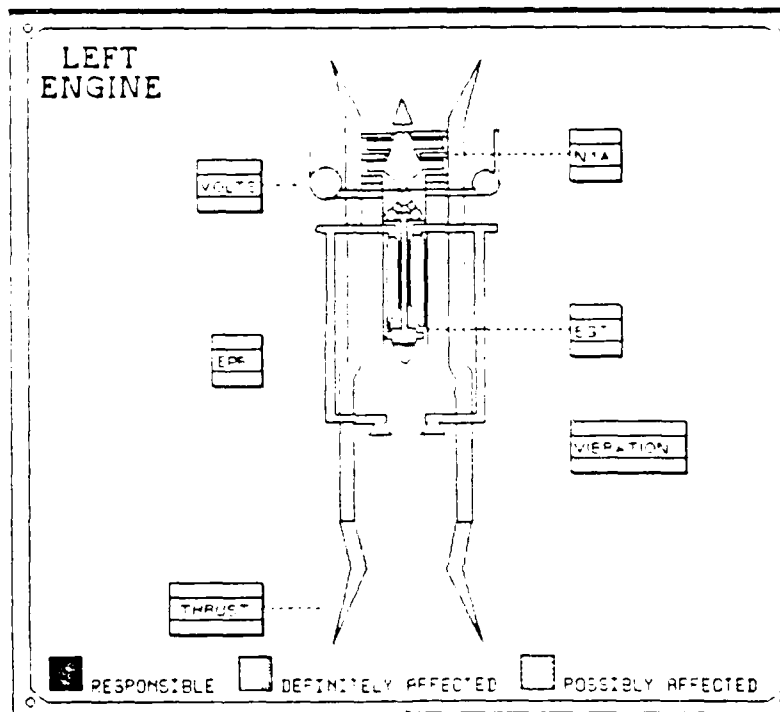


Figure 20. Left Engine Display

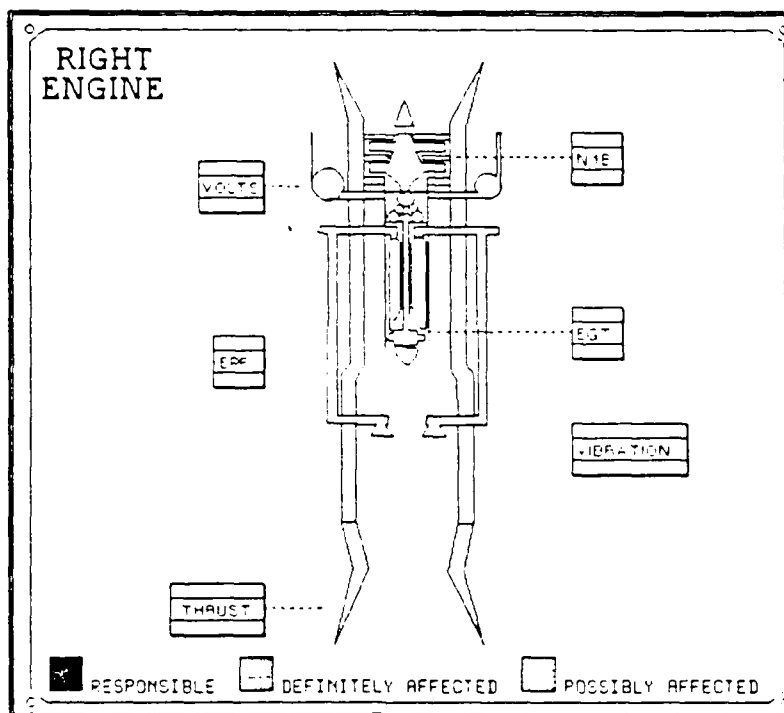


Figure 21. Right Engine Display.

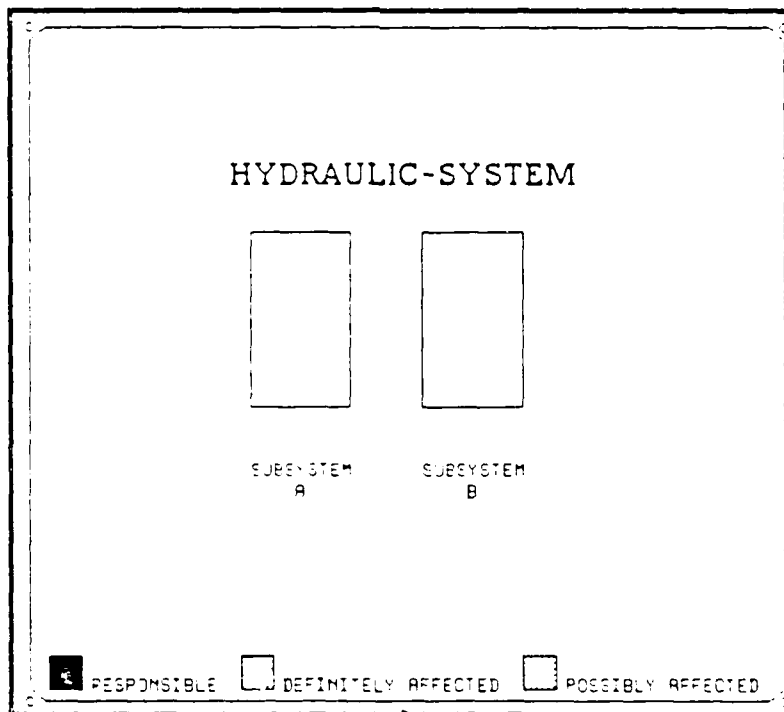


Figure 22. Hydraulic System Display.

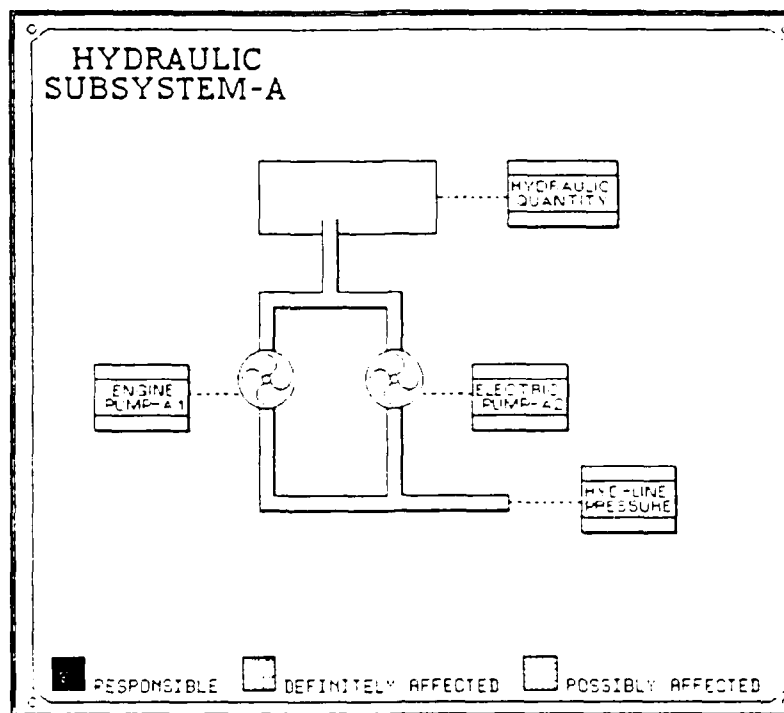


Figure 23. Left Hydraulic Subsystem Display.

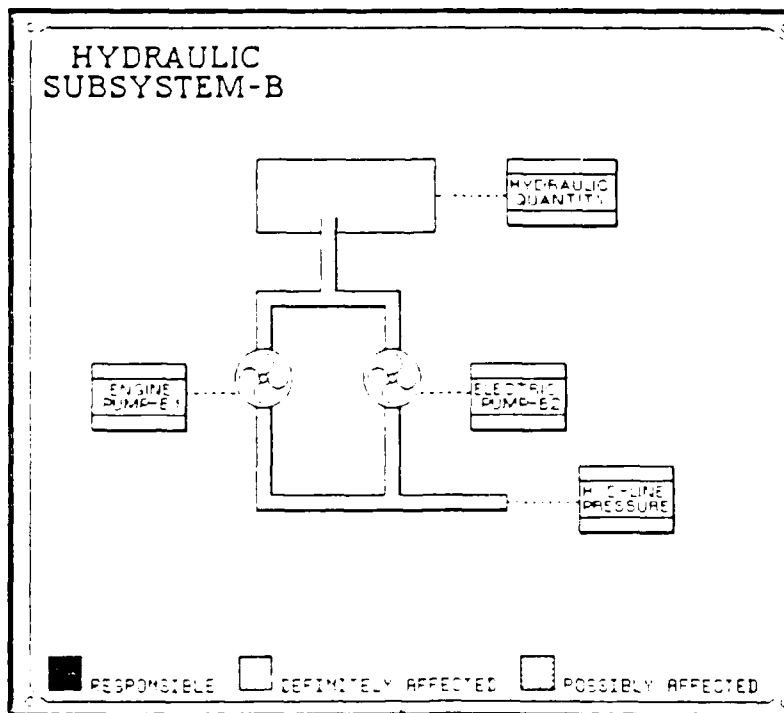


Figure 24. Right Hydraulic Subsystem Display.

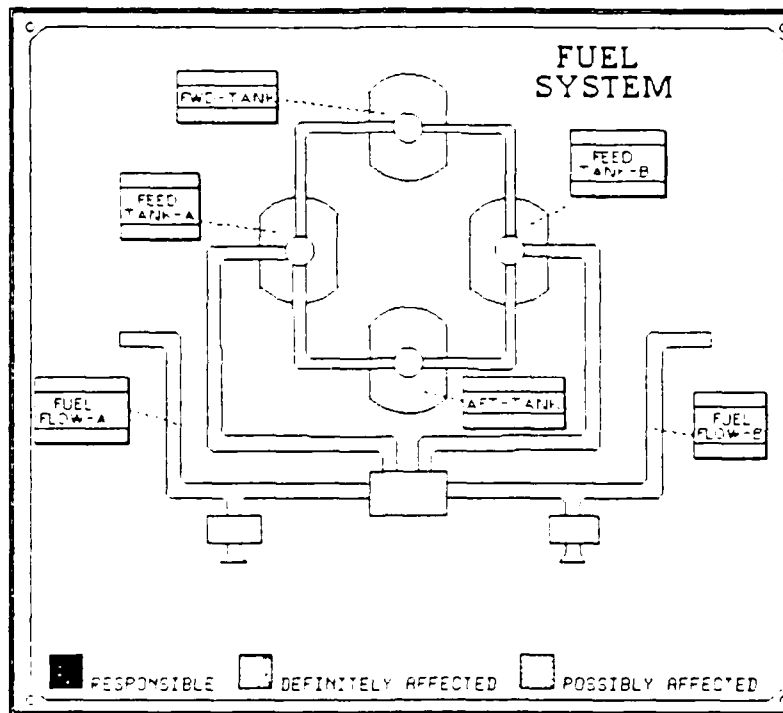


Figure 25. Fuel System Display.

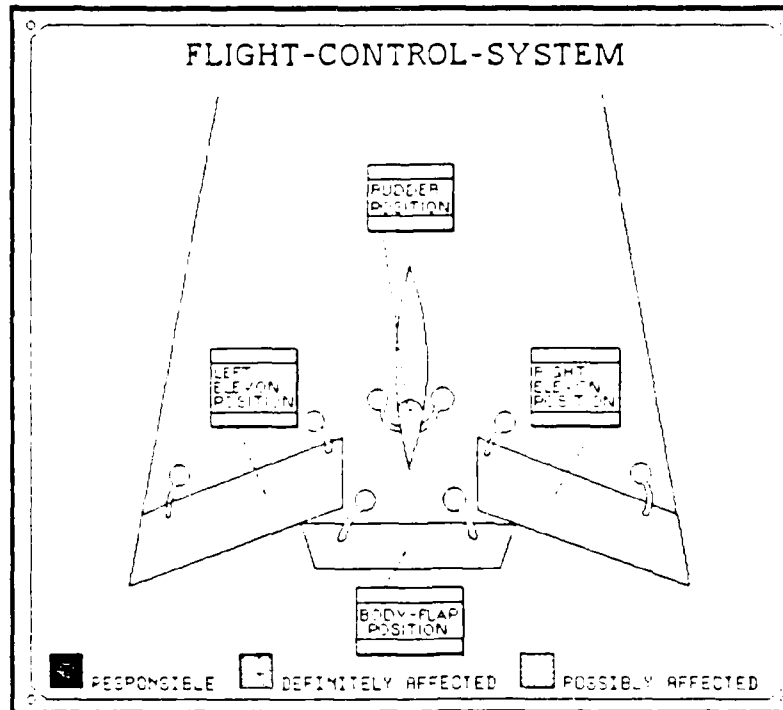


Figure 26. Flight Control System Display.

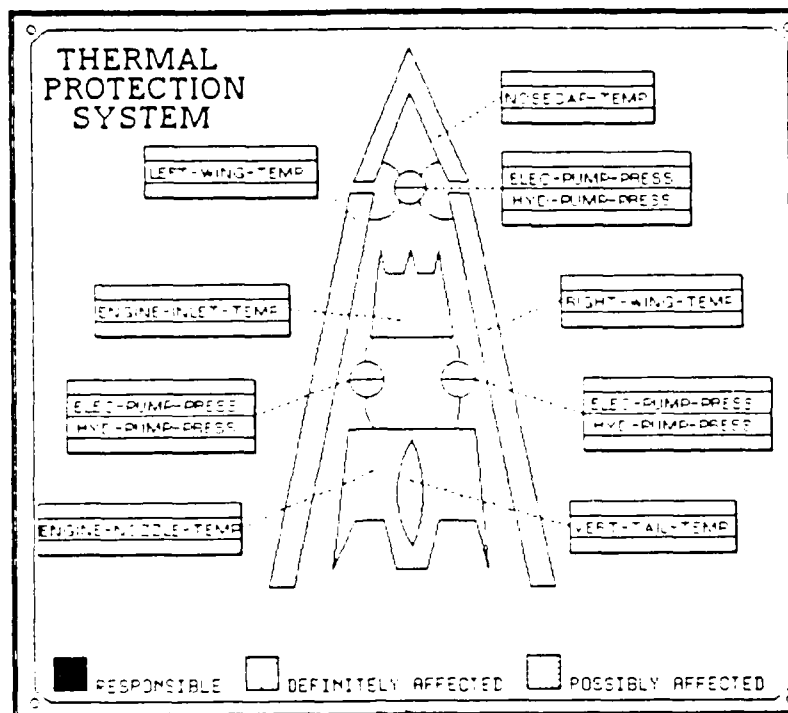


Figure 27. Thermal Protection System Display.

symptoms. If the associated sensor has a "causes" link in the knowledge base, the remediation function looks for a "causes" link whose result will counteract the symptom that the sensor is reporting. For example, if the symptom is "Airspeed Low", then the remediation function will look for a "causes" link that says "<some action> causes Increase Airspeed."

Once the highest-level fault symptom and an appropriate counter-action are found, the remediation function will work backwards from the "action" part of the "causes" link until it finds the most elementary action that will eventually produce the desired counter-action to the highest-level fault symptom. This process is repeated for all unique sequences of actions that will produce the same fault symptom counter-action.

Next, the group of action sequences must be pruned. This is done by keeping only those action sequences that counteract the greatest number of fault symptoms. For example, assume there are three current fault symptoms.

1. Airspeed Low
2. Fuel-Pressure Low
3. Thrust Low

and there are two competing remedial action sequences,

1. Engage Afterburner causes Increase Thrust.
2. Increase Thrust causes Increase Airspeed

and,

1. Decrease Weight causes Decrease Drag,
2. Decrease Drag causes Increase Airspeed.

In this case, the first action sequence would be preferred, because it counteracts two of the three current symptoms, whereas the second action sequence only counteracts the "Airspeed Low" symptom.

Remedies are displayed in textual form on the diagnosis pane of the display. Figure 28 shows an example remedy listing.

REMEDY 1 OF 1

REMEDIAL ACTION

INCREASE BOOST-PUMP-B-PRESSURE
INCREASE FUEL-FLOWB
INCREASE THRUSTB
INCREASE TOTAL-THRUST
INCREASE MACHA

Diagnosis pane

Figure 28. Sample Remedy List.

V. Results

To demonstrate its functions and capabilities, the NASP System Status Monitor was presented with three different types of test inputs. The first series of tests involved a set of four logically related fault symptoms. These symptoms were entered interactively into the SSM five times. On the first trial, all four symptoms were entered. For the second through fifth trials, a different one of the four symptoms was omitted.

The second series of test inputs also included a set of four logically related fault symptoms. Again, these symptoms were entered five times, with all four symptoms entered on the first trial. For the second through fifth trial, a single additional symptom was added to the other four.

The results of the test runs show that the NASP SSM will successfully diagnose sets of logically related fault symptoms, using both fault association rules and functional relationship fault hypothesis generation. However, if the symptoms are somehow discontinuous, or random and unrelated, the best the SSM can do is to produce a default hypothesis.

5.1 Test 1A

The first set of fault symptoms used to test the performance of the NASP System Status Monitor included logically related faults. The intent was that a fault in one of the hydraulics subsystems would affect the flight control system, with the flight control anomaly ultimately impairing overall aircraft performance. The following four fault symptoms were given to the SSM to perform this test;

1. Hyd-SubsysA-Pressure - Low
2. Hyd-PumpA1-Pressure - Low
3. Left-Elevon-Position - Low
4. Climb-RateA - Low

When presented with these four faults, the SSM did not produce a Stage 1 diagnosis, but returned two Stage 2 hypotheses. Both of these hypotheses contained two remedies.

Figure 29 is a diagram of the fault propagation path produced by the SSM for the first Stage 2 hypothesis. In this hypothesis, the gearbox in the left engine is the responsible component, and the fault effects propagate directly to the mission level at the top of the functional hierarchy. The legend at the bottom of Figure 29 shows which components in the diagram are responsible components, definitely affected, possibly affected, or sensors.

One interesting aspect of this hypothesis is its apparent deviation from the intent of its test set of fault symptoms. The fault propagation was intended to begin with a fault in or near the left engine-driven hydraulic pump (Engine-Hyd-PumpA). This fault was supposed to propagate through the left hydraulic subsystem to the flight control system, where it would affect the hydraulically-driven control surface actuators. The affected actuators would incorrectly position a control surface which would aerodynamically impair the climb rate. Figure 29 appears to show that the fault propagation within the flight control system actually has no effect on the upward propagation of the fault in the functional hierarchy. This is true to a point, since the propagation path just as easily could have gone from "Flight-Control-SysA" to "Takeoff-Plane" as it did from "Hydraulic-SystemA" to "Takeoff-Plane." The Takeoff-Plane (and all other instances of the "Plane") are functionally dependent on both the Flight Control System and the Hydraulic System. The reason one path was chosen over the other lies in the knowledge base. The "functional-dependents" links for the "Hydraulic-LineA" are ordered unintentionally so that the "Hydraulic-SubsystemA" comes before the "Left-Elevon-Actuator-1." Therefore, the propagation path through the hydraulic system is explored (and found to lead to the top of the functional hierarchy) before the path through the Flight Control System is attempted.

The second hypothesis for this set of fault symptoms is the same as the first except that the left engine-driven hydraulic pump is now the responsible component (see Figure 30). Since the

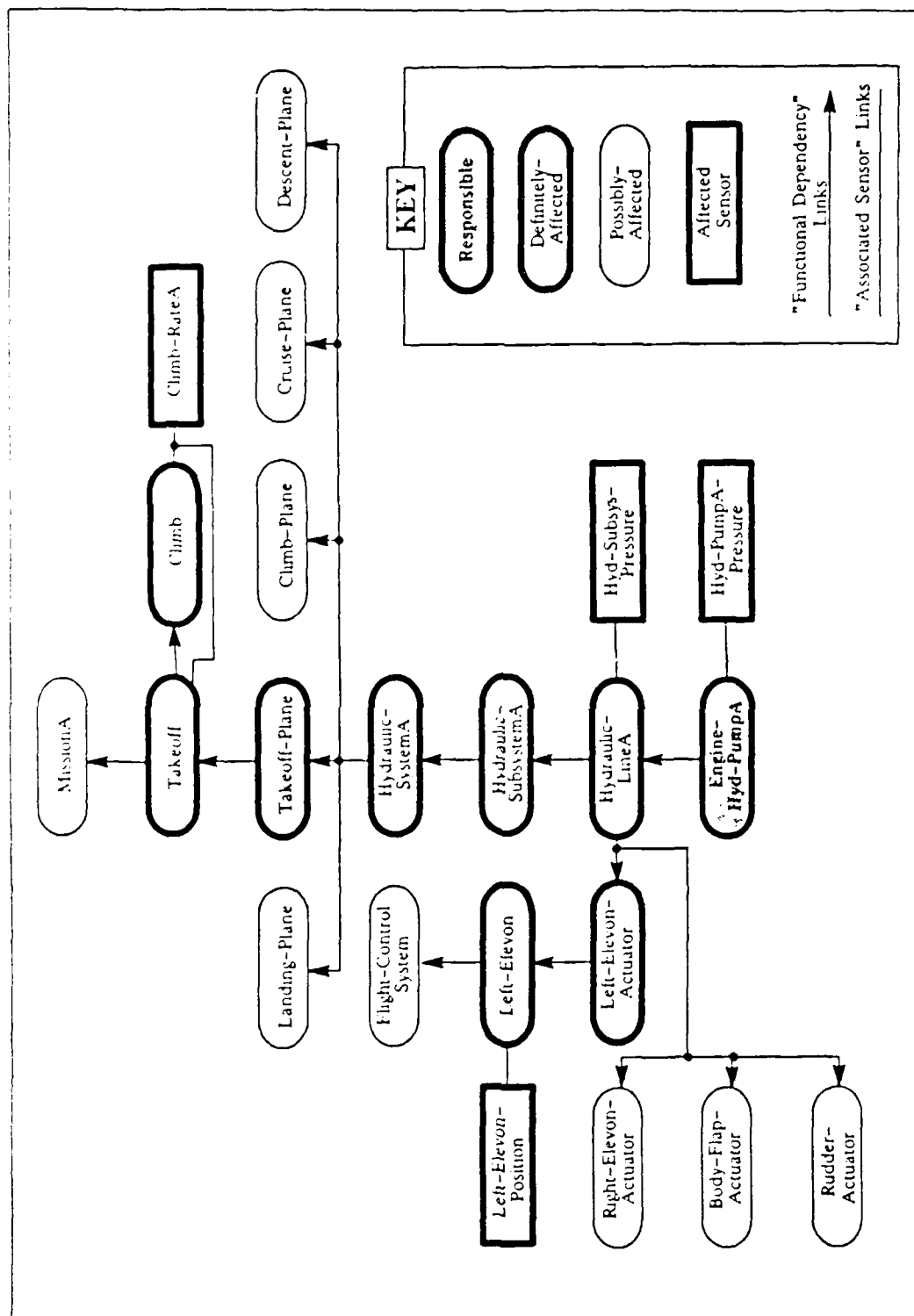


Figure 29 Test 1A Fault Tree Diagram (first hypothesis).

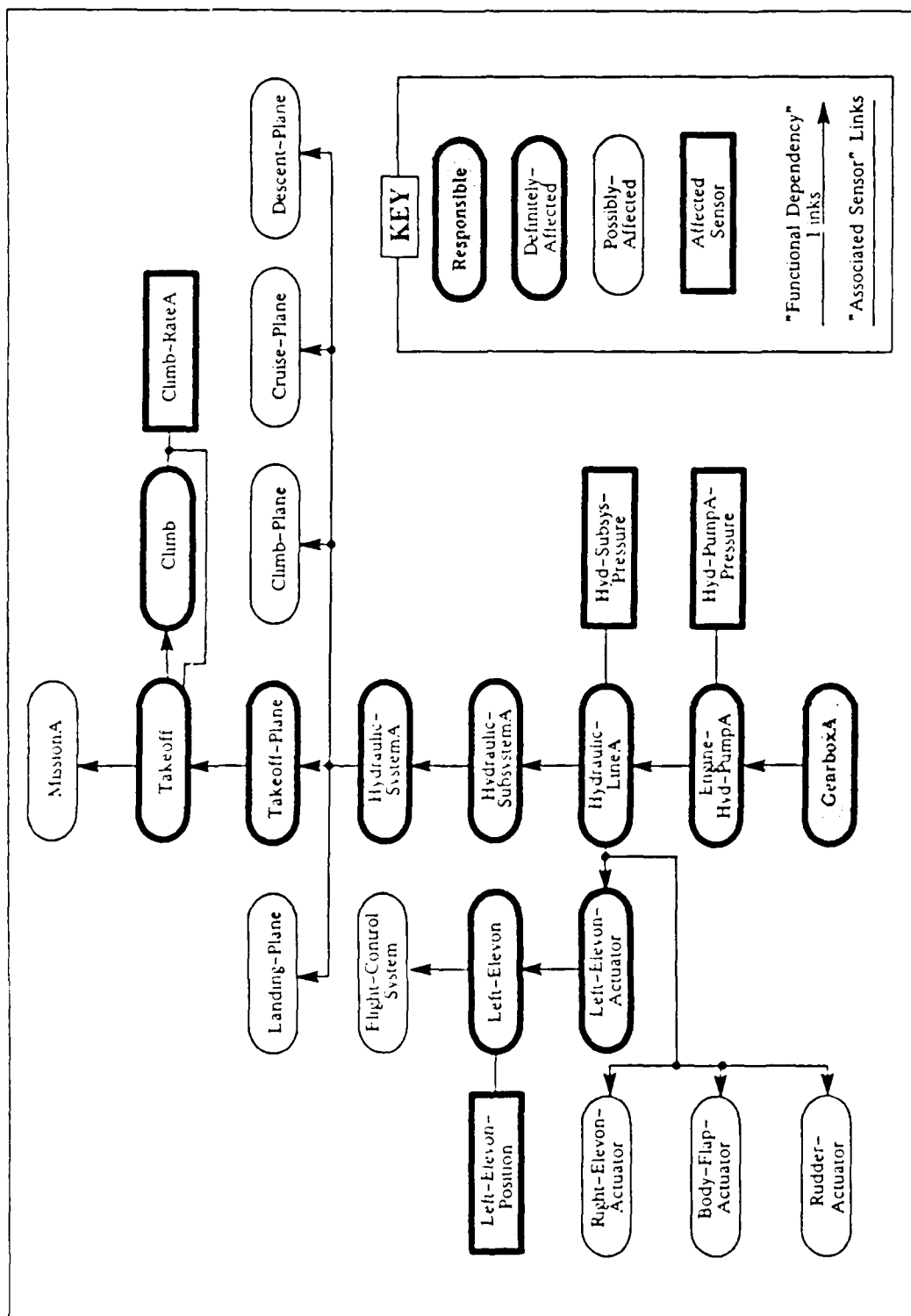


Figure 30. Test 1A Fault Tree Diagram (second hypothesis).

REMEDY 1 OF 2

REMEDIAL ACTION
MOVE-RIGHT CONTROL-STICK
DECREASE LEFT-ELEVON-POSITION
DECREASE DRAG
INCREASE MACHA
INCREASE CLIMB-RATEA

Diagnosis pane

REMEDY 2 OF 2

REMEDIAL ACTION
MOVE-RIGHT CONTROL-STICK
DECREASE LEFT-ELEVON-POSITION
DECREASE DRAG
INCREASE AIRSPEEDA
INCREASE CLIMB-RATEA

Diagnosis pane

Figure 31. Test 1A Remedies.

gearbox has no associated sensor, Stage 2 could not determine if a gearbox fault or a hydraulic pump fault was causing the "Hyd-PumpA1-Pressure" symptom. Therefore, Stage 2 produced a hypothesis for each of these possibilities.

Both hypotheses for this test have the same two remedies, shown in Figure 31. These remedies are directed towards the highest-level fault symptom, the low "Climb-RateA." Also, both remedies counteract two of the four fault symptoms ("Climb-RateA." and "Left-Elevon-Position").

4.4 Test 1B

This test was the same as Test 1A except that the "Left-Elevon-Position" symptom was deleted. The effect of this deletion on the fault hypotheses is not significant, but the absence of the

"Left-Elevon-Position" symptom has a drastic effect on the suggested remedial actions.

Figure 32 shows the fault propagation path for Test 1B. This diagram is the same as for Test 1A, except that the fault effects for the entire Flight Control System are omitted. In this test case, the beginning and end of the propagation path are established by the "Hyd-PumpA1-Pressure" and "Climb-RateA" symptoms, respectively. The three components in the center of the propagation path (Hydraulic-SubsystemA, Hydraulic-SystemA, and Takeoff-Plane) do not have associated sensors, and so just carry the propagation. They do not have the potential to stop the propagation.

Let us assume that either the Hydraulic-SubsystemA or Hydraulic-SystemA had their own sensors, and those sensors were unaffected by the current symptoms. In this case, they could stop the fault propagation and Stage 2 would be forced to seek another path, such as through the Flight Control System. Under this assumption, Test 1 would still produce the same basic hypotheses, with the propagation path going from the "Flight-Control-SysA" rather than from the "Hydraulic-SystemA" to the "Takeoff-Plane." The same could not be said for the hypotheses produced in Test 1B. Here the hydraulic subsystem symptoms would be cut off from the higher level symptom, forming two "islands" of symptoms and their effects. These "islands" are in fact how a default hypothesis is represented when no valid hypothesis can be generated.

The major difference between Test 1A and Test 1B is in the generation of remedial actions. The two Test 1A hypotheses each had two remedies, and each remedy counteracted two symptoms (Climb-RateA and Left-Elevon-Position). Since "Left-Elevon-Position" is no longer a symptom, it cannot affect the choice of remedial actions. In Test 1B, the best remedies to be found counteract only one symptom, and there are 22 such remedies, shown in Figure 33.

Clearly, this is a case where some other criteria must be used to select a smaller number of appropriate remedies.

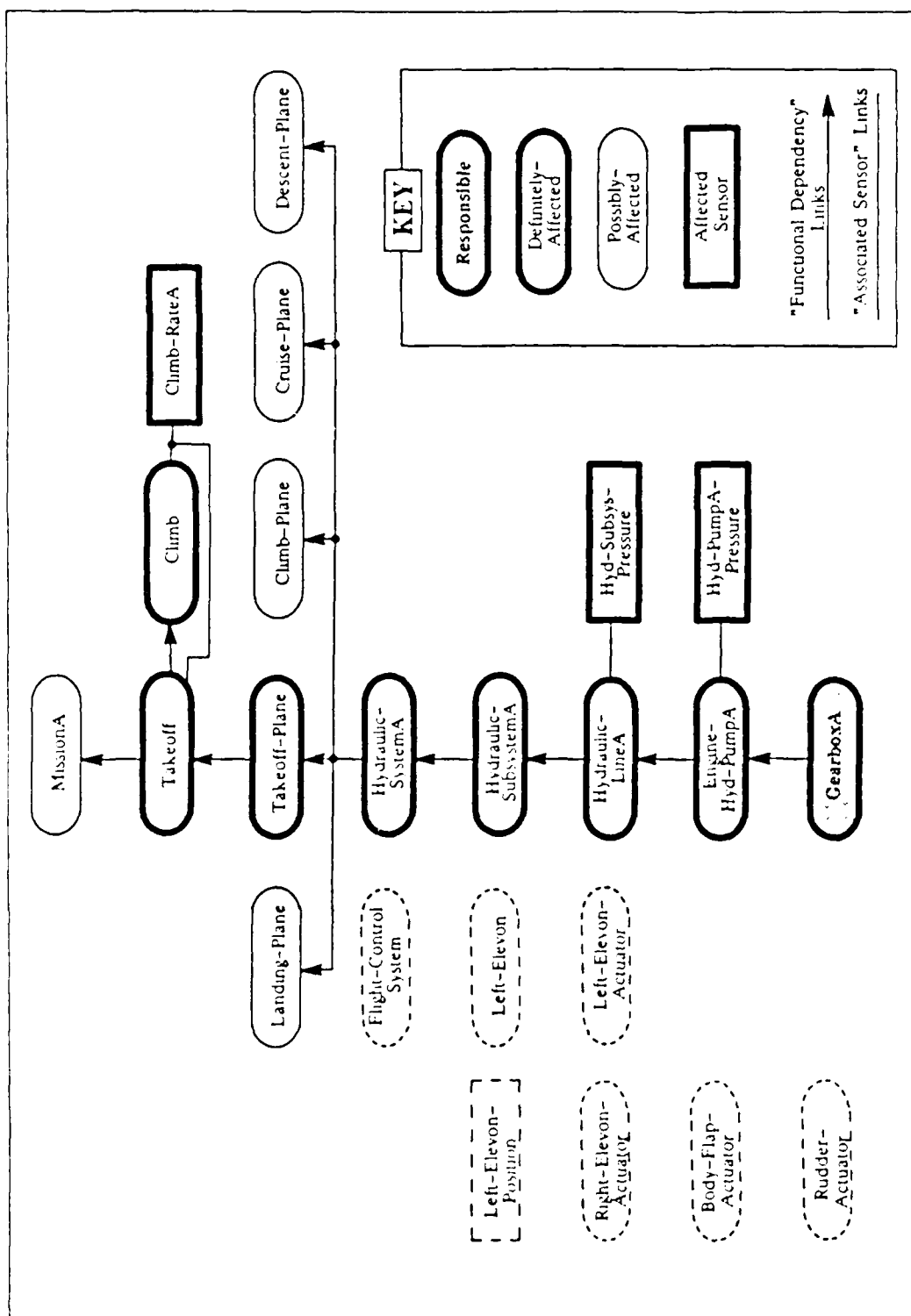


Figure 32. Test 1B Fault Tree Diagram.

```

(DECREASE TOTAL-FUEL-QUANTITY)
(DECREASE WEIGHT)
(INCREASE CLIMB-RATEA)

(MOVE-AFT CONTROL-STICK)
(INCREASE BODY-FLAP-POSITION)
(INCREASE PITCHA)
(INCREASE CLIMB-RATEA)

(MOVE-FORWARD CONTROL-STICK)
(DECREASE BODY-FLAP-POSITION)
(DECREASE PITCHA)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(PUSH LEFT-RUDDER-PEDAL)
(DECREASE RUDDER-POSITION)
(DECREASE DRAG)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(MOVE-FORWARD CONTROL-STICK)
(DECREASE BODY-FLAP-POSITION)
(DECREASE DRAG)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(MOVE-LEFT CONTROL-STICK)
(DECREASE RIGHT-ELEVON-POSITION)
(DECREASE DRAG)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(MOVE-RIGHT CONTROL-STICK)
(DECREASE LEFT-ELEVON-POSITION)
(DECREASE DRAG)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(DECREASE TOTAL-FUEL-QUANTITY)
(DECREASE WEIGHT)
(DECREASE DRAG)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

```

Figure 33. Test 1B Remedies.

(INCREASE BOOST-PUMPB-PRESSURE)
(INCREASE FUEL-FLOWB)
(INCREASE THRUSTB)
(INCREASE TOTAL-THRUST)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(INCREASE N1B)
(INCREASE EPRB)
(INCREASE THRUSTB)
(INCREASE TOTAL-THRUST)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(INCREASE BOOST-PUMPA-PRESSURE)
(INCREASE FUEL-FLOWA)
(INCREASE THRUSTA)
(INCREASE TOTAL-THRUST)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(INCREASE N1A)
(INCREASE EPRA)
(INCREASE THRUSTA)
(INCREASE TOTAL-THRUST)
(INCREASE MACHA)
(INCREASE CLIMB-RATEA)

(MOVE-FORWARD CONTROL-STICK)
(DECREASE BODY-FLAP-POSITION)
(DECREASE PITCHA)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(PUSH LEFT-RUDDER-PEDAL)
(DECREASE RUDDER-POSITION)
(DECREASE DRAG)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(MOVE-FORWARD CONTROL-STICK)
(DECREASE BODY-FLAP-POSITION)
(DECREASE DRAG)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

Figure 33 Test 1B Remedies (continued)

(MOVE-LEFT CONTROL-STICK)
(DECREASE RIGHT-ELEVON-POSITION)
(DECREASE DRAG)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(MOVE-RIGHT CONTROL-STICK)
(DECREASE LEFT-ELEVON-POSITION)
(DECREASE DRAG)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(DECREASE TOTAL-FUEL-QUANTITY)
(DECREASE WEIGHT)
(DECREASE DRAG)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(INCREASE BOOST-PUMPB-PRESSURE)
(INCREASE FUEL-FLOWB)
(INCREASE THRUSTB)
(INCREASE TOTAL-THRUST)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(INCREASE N1B)
(INCREASE EPRB)
(INCREASE THRUSTB)
(INCREASE TOTAL-THRUST)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(INCREASE BOOST-PUMPA-PRESSURE)
(INCREASE FUEL-FLOWA)
(INCREASE THRUSTA)
(INCREASE TOTAL-THRUST)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

(INCREASE N1A)
(INCREASE EPRA)
(INCREASE THRUSTA)
(INCREASE TOTAL-THRUST)
(INCREASE AIRSPEEDA)
(INCREASE CLIMB-RATEA)

Figure 33 Test 1B Remedies (continued)

5.3 Test 1C

This test was the same as Test 1A except that the "Climb-RateA" symptom was deleted. This deletion changed the entire outcome of the diagnosis process, since no valid hypothesis could be generated. The result is a default hypothesis, where each sensor is declared to be "definitely affected" and all components on what would otherwise be called the propagation path are declared to be "possibly affected." Because the affected components cannot be connected to form a single propagation path, the default hypothesis has "islands" of fault effects which can be seen in Figure 34.

Since Stage 2 of the diagnosis function did not produce a valid fault hypothesis, the remediation function did not produce any remedies. Although the current implementation of the SSM will not attempt to produce remedies unless there is at least one valid hypothesis, there is no conceptual prohibition to doing so. The assumed intent of the remediation function was to counteract the effects of as many fault symptoms as possible without regard for the cause of those symptoms. Under that assumption, the absence of a diagnosis hypothesis should not preclude an attempt to counteract the fault symptoms. Therefore, the remediation should perhaps be modified to recommend remedial action in all cases.

This test shows that the deletion of a single symptom can prevent the generation of a valid hypothesis. If Stage 2 could recognize the absence of a key symptom, it may be able to compensate and produce a valid hypothesis.

5.4 Test 1D

This test was the same as Test 1A except that the "Climb-RateA" symptom was deleted. This deletion prevented Stage 2 of the diagnosis function from predicting any possible effect of the fault situation on the highest level of the functional hierarchy. The remainder of the diagnosis is the same as Test 1A, as can be seen in diagram in Figure 35.

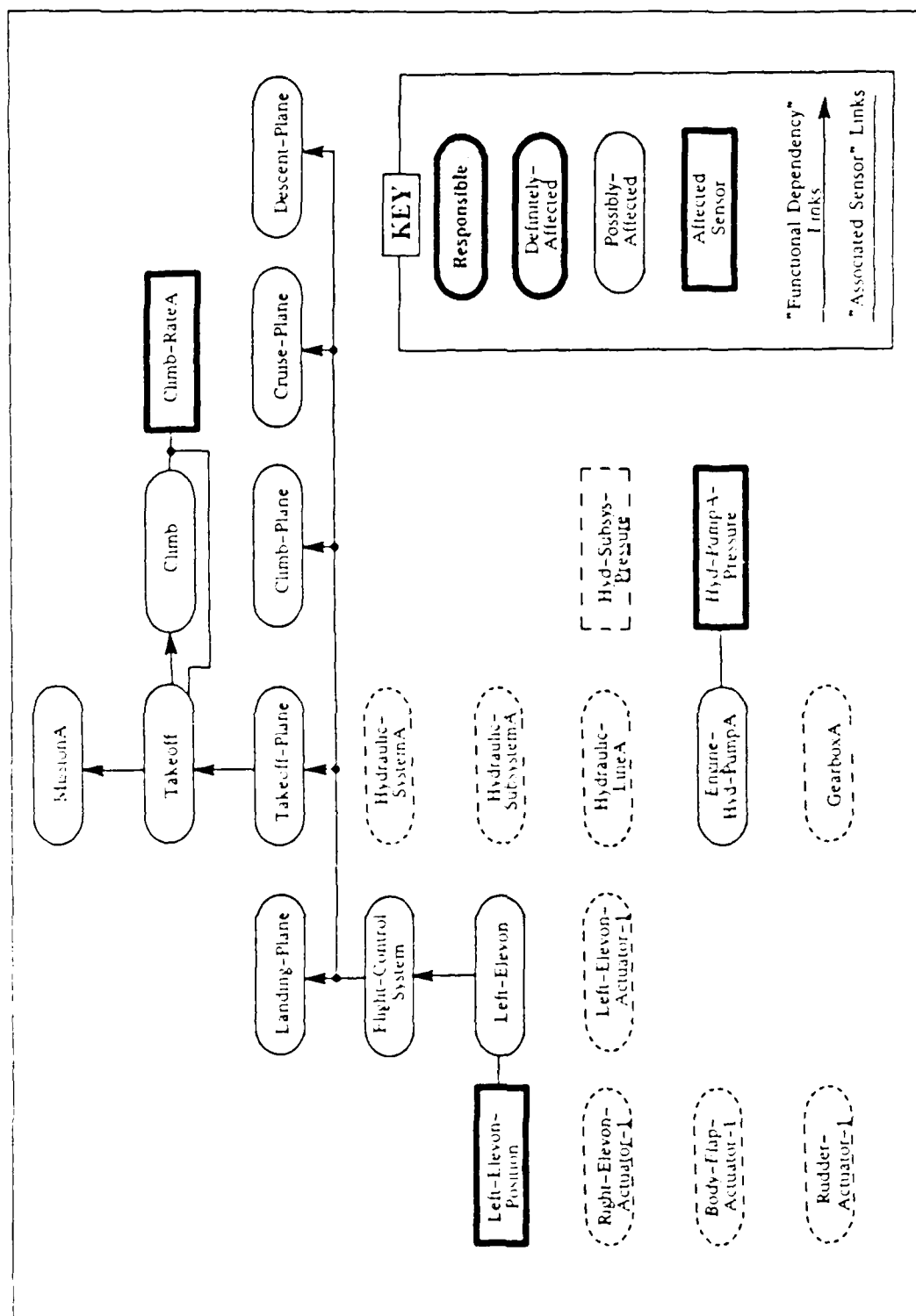


Figure 34 Test 1C Fault Tree Diagram

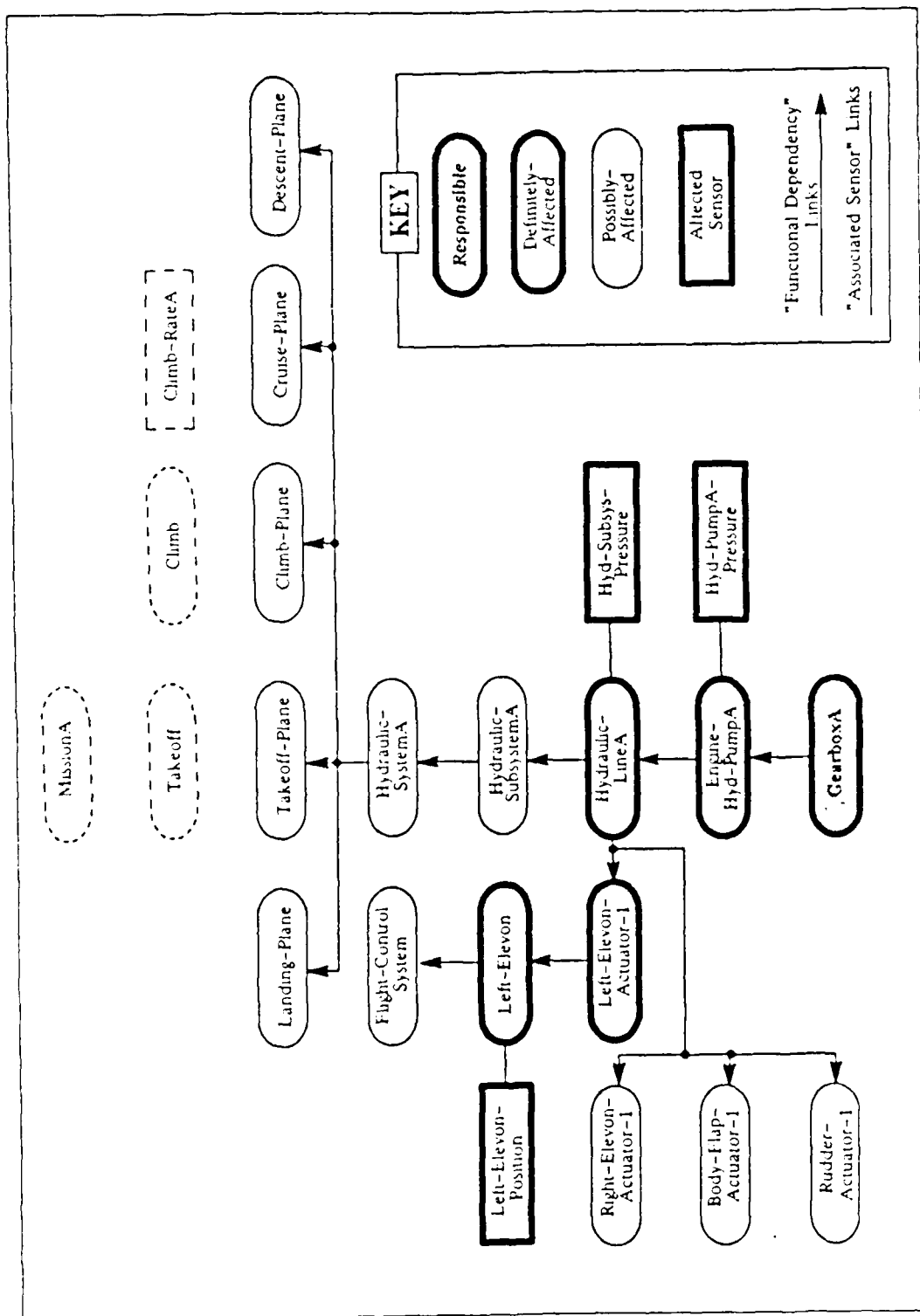


Figure 35 Test ID Fault Tree Diagram.

REMEDY 1 OF 1

REMEDIAL ACTION

MOVE-LEFT-CONTROL-STICK
INCREASE-LEFT-ELEVON-POSITION

Diagnosis pane

Figure 36. Test 1D Remedy.

The deletion of the "Climb-RateA" symptom also had an effect on the remedies associated with this diagnosis. The remediation function attempts to counteract the effects of the highest-level fault symptom. In all the previous test cases, the low "Climb-RateA" was the highest-level symptom. Since it is not present in this test, the remediation function chose the next highest symptom, the low "Left-Elevon-Position." For both hypotheses in this test case, the same remedial action sequence was prescribed. The recommended remedy for this test is shown in Figure 36.

5.5 Test 1E

This test was the same as Test 1A except that the "Hyd-PumpA1-Pressure" symptom was deleted. This change had the effect of removing what was the beginning of the fault propagation path in the previous test cases, as shown in Figure 37. This set of symptoms still produced a valid hypothesis because "Hydraulic-LineA" is also a primitive component and can therefore serve as the beginning of a valid propagation path. The two remedial actions proposed for this test case are the same as in Test 1A (see Figure 31).

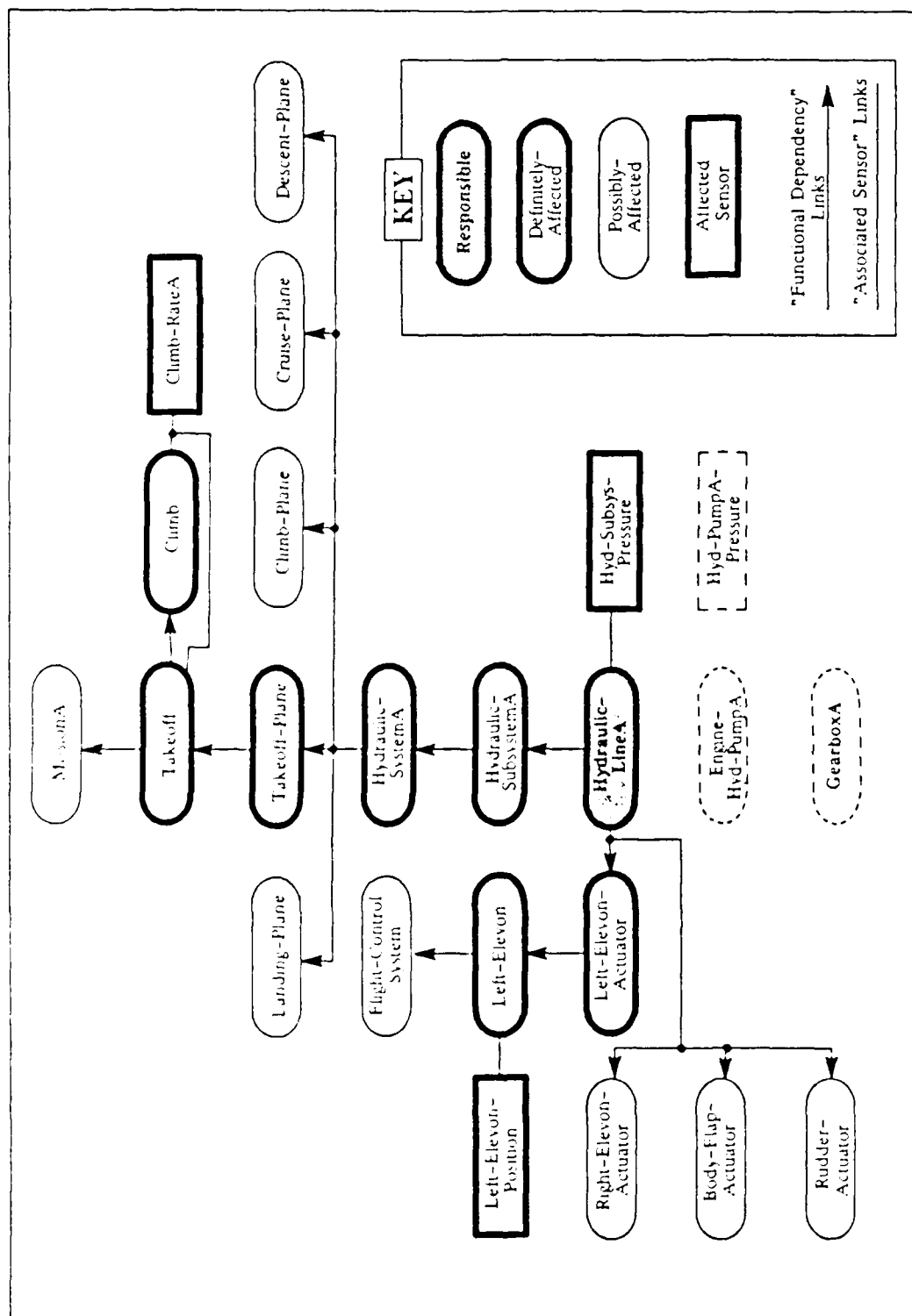


Figure 37. Test 1E Fault Tree Diagram.

5.6 Test 2A

Test 2A involved another set of fault symptoms that were logically related and therefore should have produced a valid fault hypothesis. This set of symptoms was intended to show the propagation of a fault in the fuel system to one of the engines, and the effect of the engine problem manifesting itself in a low Mach number. The set of fault symptoms for this test were:

1. MachA - low
2. ThrustB - low
3. Fuel-FlowB - high
4. Feed-TankB-Quantity - low.

As in Test 1A, the fault symptoms in Test 2A did not produce a Stage 1 diagnosis, but they did produce three Stage 2 diagnoses. These diagnoses are shown in the fault propagation diagrams of Figures 38, 39, and 40.

The fault propagation diagram in Figure 38 shows that the fault symptoms for this test did produce the intended propagation path. The only difference in the three hypotheses is the responsible component. The "EngineB-Feed-Tank" is the lowest-level component that must be in the propagation path, because it is the lowest-level component with an affected sensor. The "EngineB-Feed-Tank" is functionally dependent on both the "Fwd-Tank-Transfer-Pump" and the "Aft-Tank-Transfer-Pump." Neither of these components have associated sensors, so they can also be considered responsible components in the second and third hypotheses.

For this set of fault symptoms, Stage 2 produces the same single remedy for each of the three hypotheses. This remedy attempts to counteract the highest-level fault symptom, low "MachA." The remedy, shown in Figure 41, actually counteracts two other fault symptoms in addition to "MachA."

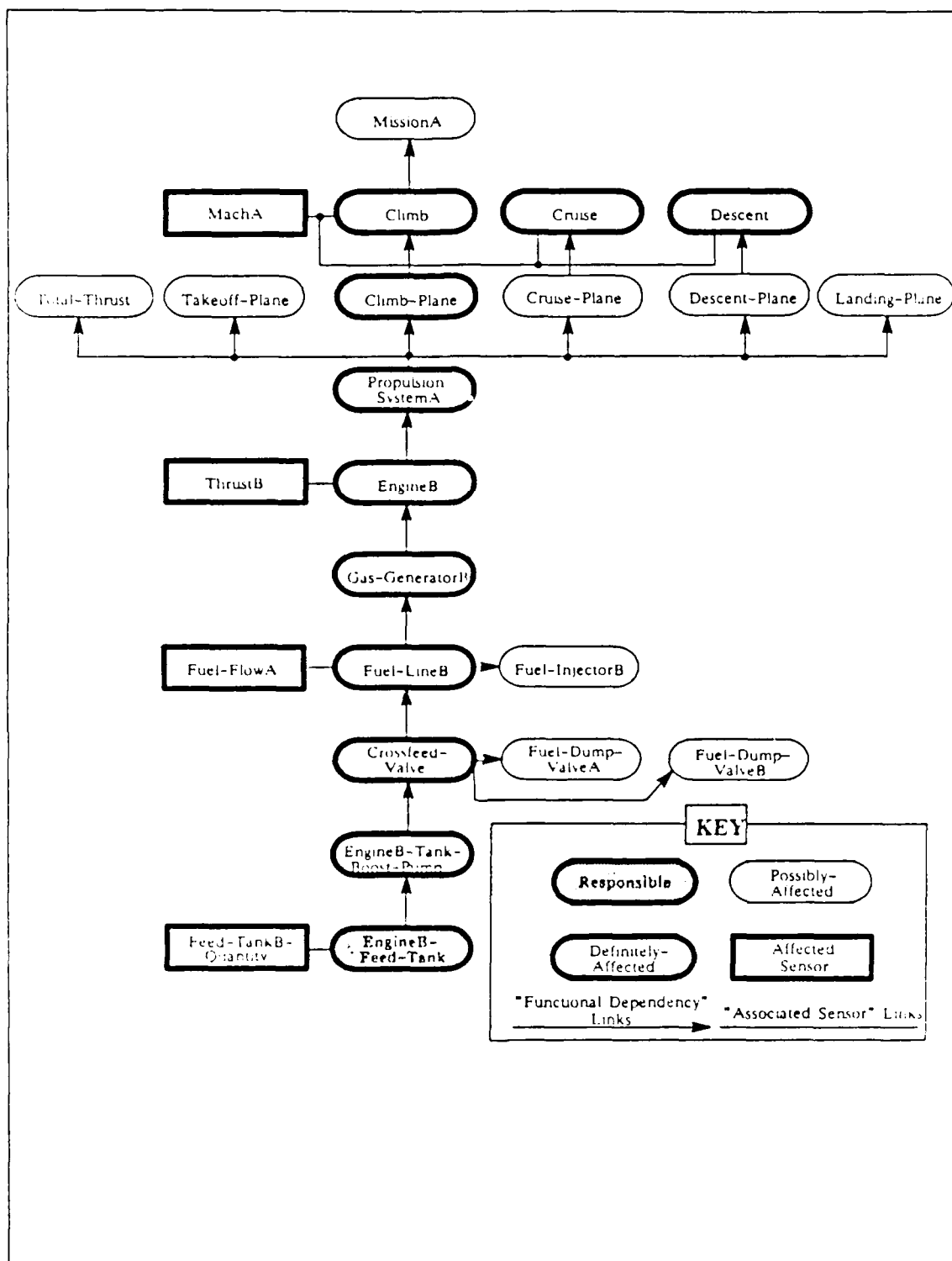


Figure 38. Test 2A Fault Tree Diagram (first hypothesis).

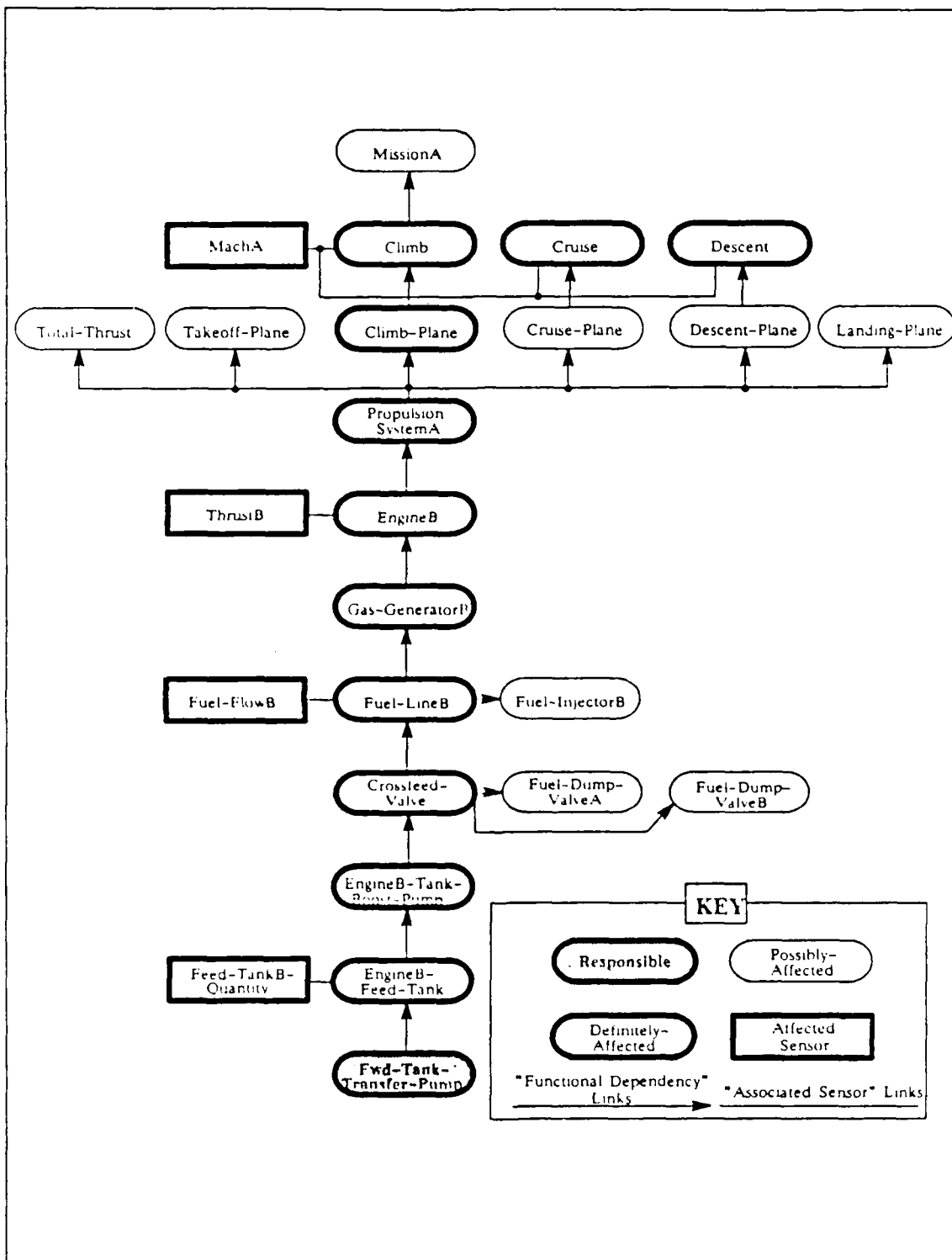


Figure 39. Test 2A Fault Tree Diagram (second hypothesis).

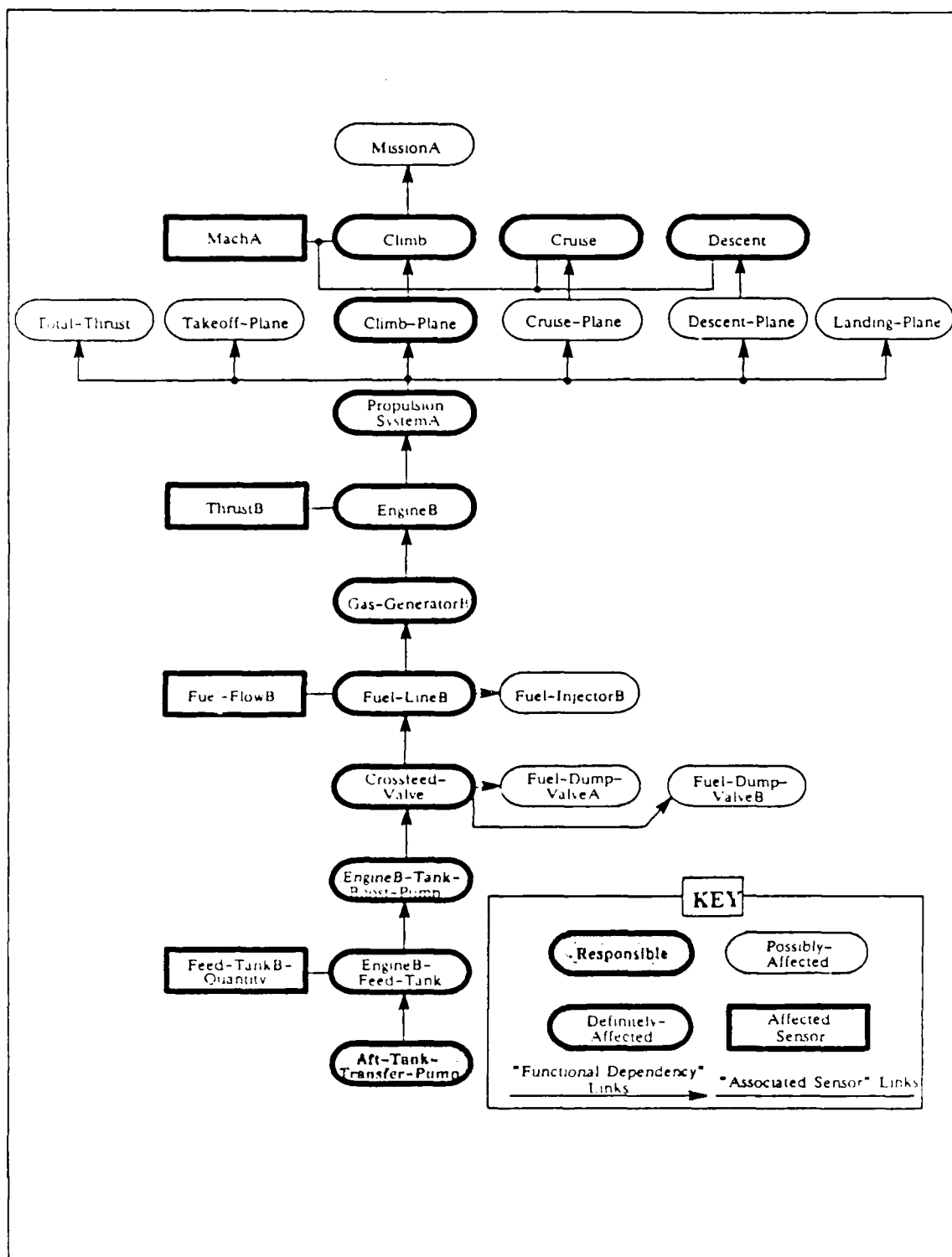


Figure 40. Test 2A Fault Tree Diagram (third hypothesis)

REMEDY 1 OF 1 REMEDIAL ACTION INCREASE BOOST-PUMPB-PRESSURE INCREASE FUEL-FLOWB INCREASE THRUSTB INCREASE TOTAL-THRUST INCREASE MACHA Diagnosis pane
--

Figure 41. Test 2A Remedy.

5.7 Test 2B

This test is the same as Test 2A except that an additional symptom, low "Fwd-Tank-Quantity," is included. The addition of this symptom at the bottom of the fault propagation path focuses the diagnosis process. The result is a single fault hypothesis with the "Fwd-Fuel-Tank" as the responsible component. The remainder of the fault propagation path, shown in Figure 42, is the same as in Test 2A.

In this case, the additional symptom does not have an effect on the remedial action function. The same remedy is produced for the single Test 2B hypothesis as was produced for each of the Test 2A hypotheses.

5.8 Test 2C

This test is the same as Test 2A except that an additional symptom, low "N1B," is included. Because two of the current symptoms (high "Fuel-FlowB" and low "N1B") matched one of the rules in the Stage 1 knowledge base, this set of symptoms produced a Stage 1 diagnosis. The diagnosis of "Fuel-LeakB" is shown in Figure 43. Since there are three remaining symptoms not explained by the Stage 2 diagnosis, Stage 2 also attempted to produce a diagnosis.

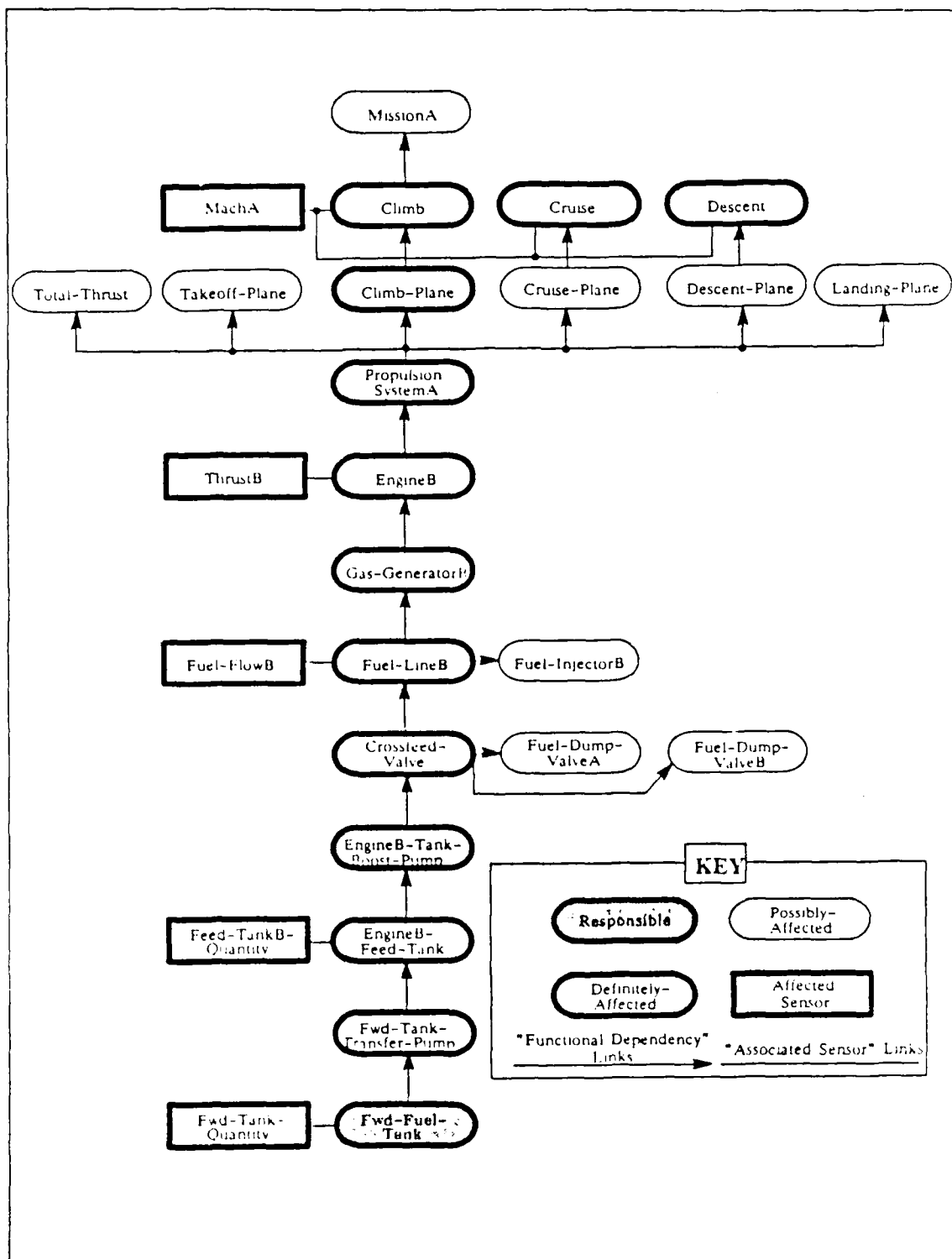


Figure 42. Test 2B Fault Tree Diagram.

CURRENT-SYMPTOMS

("MACHA" 49602 NIL "low" NIL NIL NIL)
("THRUSTB" 49606 NIL "low" NIL NIL NIL)
("FUEL-FLOWB" 49615 NIL "high" NIL NIL NIL)
("FEED-TANKB-QUANTITY" 49621 NIL "low" NIL NIL NIL)
("N1B" 49627 NIL "low" NIL NIL NIL)

STAGE-1-DIAGNOSIS

("FUELLEAKB")

STAGE-2-DIAGNOSIS

("N1B" "DEFINITELYAFFECTED")
("COMPRESSORB" "POSSIBLYAFFECTED")
("FEED-TANKB-QUANTITY" "DEFINITELYAFFECTED")
("ENGINEB-FEED-TANK" "POSSIBLYAFFECTED")
("ENGINEB-TANK-BOOST-PUMP" "POSSIBLYAFFECTED")
("CROSSFEED-VALVE" "POSSIBLYAFFECTED")
("FUEL-DUMP-VALVEA" "POSSIBLYAFFECTED")
("FUEL-DUMP-VALVEB" "POSSIBLYAFFECTED")
("FUEL-FLOWB" "DEFINITELYAFFECTED")
("FUEL-LINEB" "POSSIBLYAFFECTED")
("GAS-GENERATORB" "POSSIBLYAFFECTED")
("FUEL-INJECTORB" "POSSIBLYAFFECTED")
("THRUSTB" "DEFINITELYAFFECTED")
("ENGINEB" "POSSIBLYAFFECTED")
("PROPULSION-SYSTEMA" "POSSIBLYAFFECTED")
("TAKEOFF-PLANE" "POSSIBLYAFFECTED")
("CLIMB-PLANE" "POSSIBLYAFFECTED")
("CRUISE-PLANE" "POSSIBLYAFFECTED")
("DESCENT-PLANE" "POSSIBLYAFFECTED")
("LANDING-PLANE" "POSSIBLYAFFECTED")
("TOTAL-THRUST" "POSSIBLYAFFECTED")
("MACHA" "DEFINITELYAFFECTED")
("CLIMB" "POSSIBLYAFFECTED")
("MISSIONA" "POSSIBLYAFFECTED")
("CRUISE" "POSSIBLYAFFECTED")
("MISSIONA" "POSSIBLYAFFECTED")
("DESCENT" "POSSIBLYAFFECTED")
("MISSIONA" "POSSIBLYAFFECTED")

Figure 43. Test 2C Diagnosis.

5.9 Test 2D

This test is the same as Test 2A except that an additional symptom, low "Fwd-Elec-Cooling-Pressure," is included. This test also produces the same three basic hypotheses as Test 2A (see Figures 44, 45, and 46).

However, the figures show that the new symptom also causes the fault to begin to propagate into a different aircraft system, the Thermal Protection System.

The addition of the "Fwd-Elec-Cooling-Pressure" symptom does not alter the remedial action originally recommended in Test 2A. This tends to confirm the intuitive feeling that remedial actions taken in the Thermal Protection System would not have a direct effect on increasing Mach number.

5.10 Test 2E

This test is the same as Test 2A except that an additional symptom, low "Right-Elevon-Position," is included. However, the addition of this symptom prevents Stage 2 from producing a valid fault hypothesis. Only a default hypothesis is produced. The reason for no valid hypothesis being produced can be seen in the fault propagation path diagram in Figure 47. The two halves of the propagation path lead to the same components at the top of the functional hierarchy. However, no single path can be drawn from either half so that all five fault symptoms are traversed.

Since this is a default hypothesis, no remedial actions were recommended. As was the case with Test 1C, perhaps a remedial action for this test case would be just as appropriate and helpful as in a situation where a valid fault hypothesis was produced.

This test is the opposite of Test 1C, where the absence of a key symptom prevented generation of a valid hypothesis. Here, the presence of an extraneous symptom caused a default hypothesis. If Stage 2 could recognize the presence of the irrelevant symptom, Stage 2 may be able to ignore it and produce a valid hypothesis.

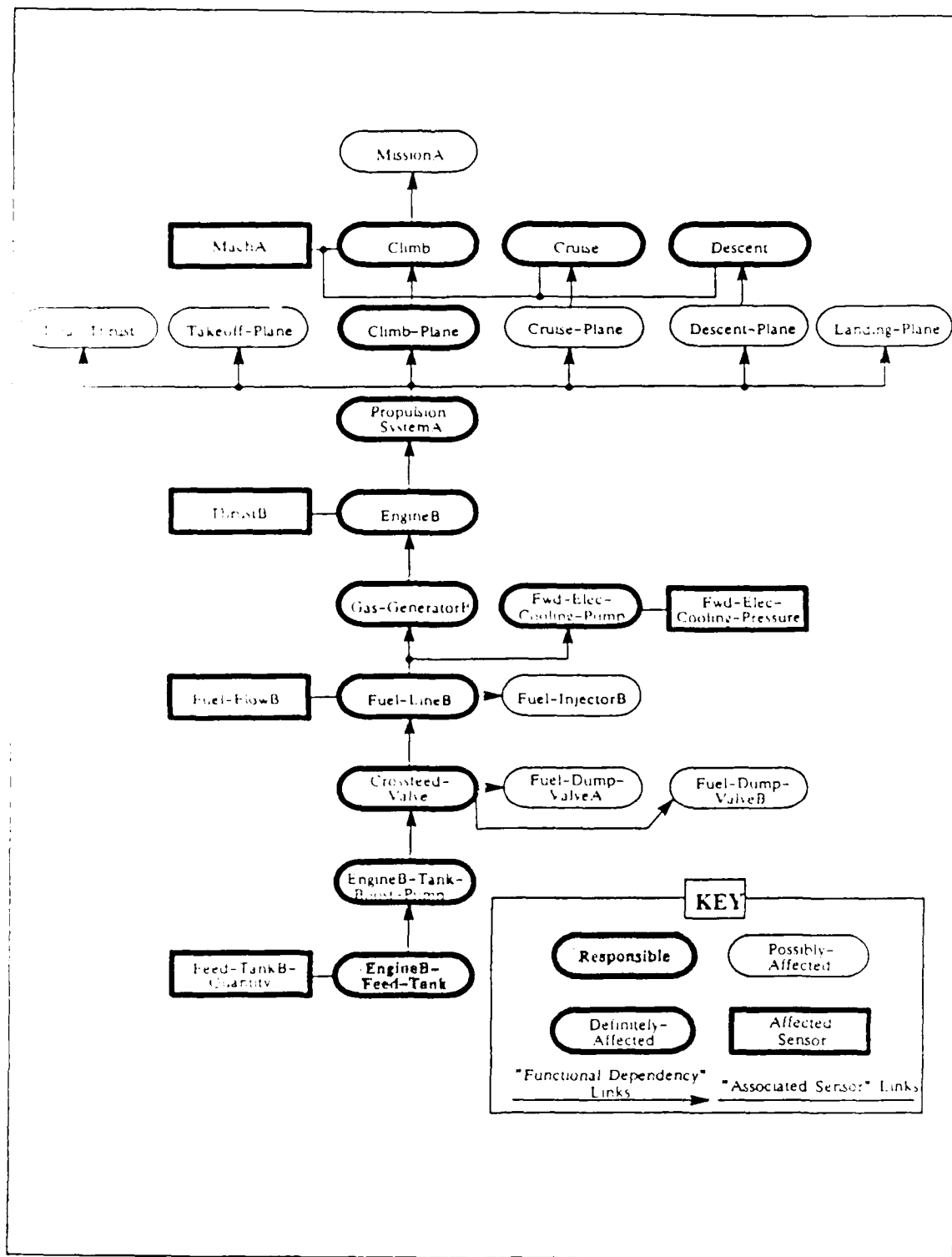


Figure 44. Test 2D Fault Tree Diagram (first hypothesis).

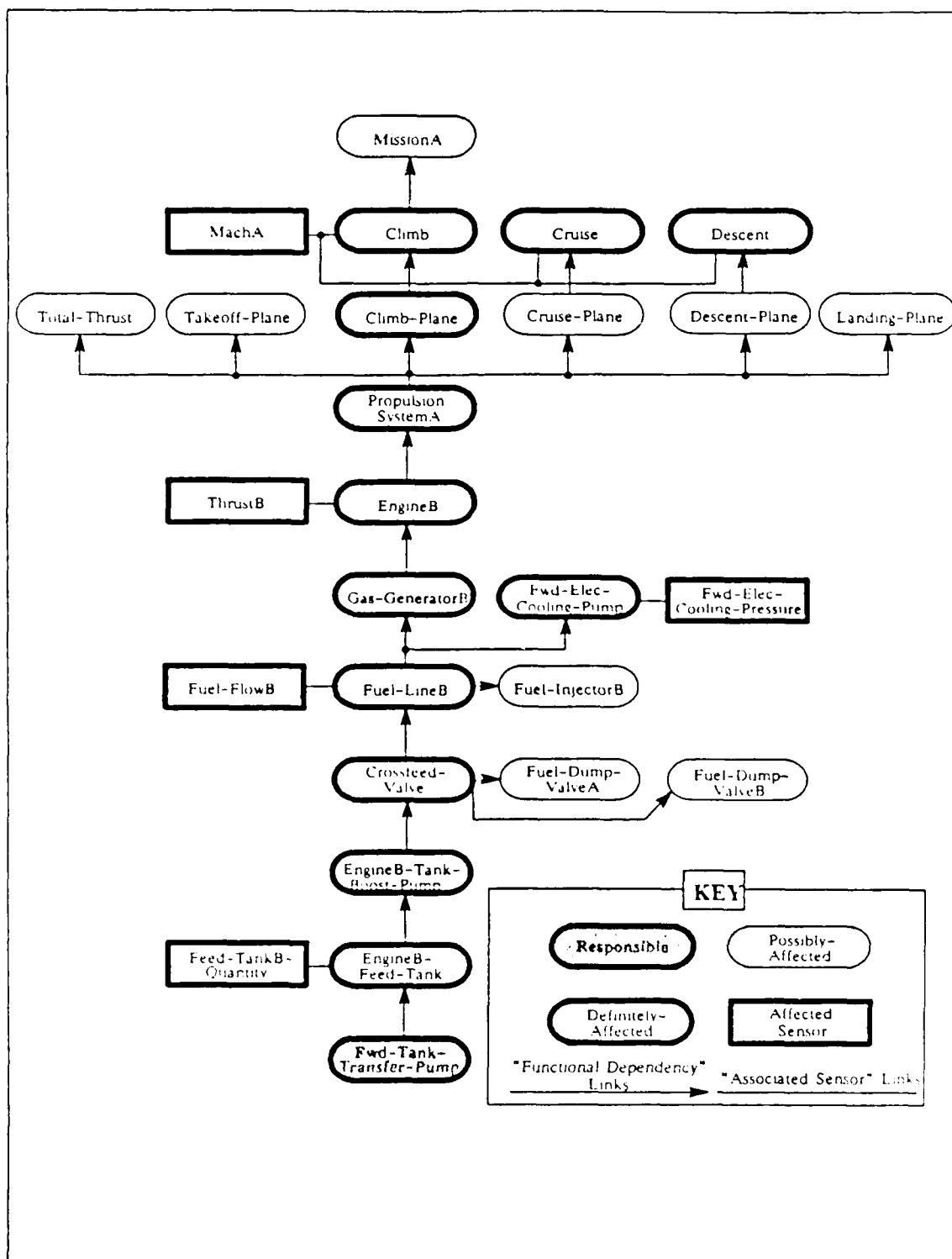


Figure 45. Test 2D Fault Tree Diagram (second hypothesis)

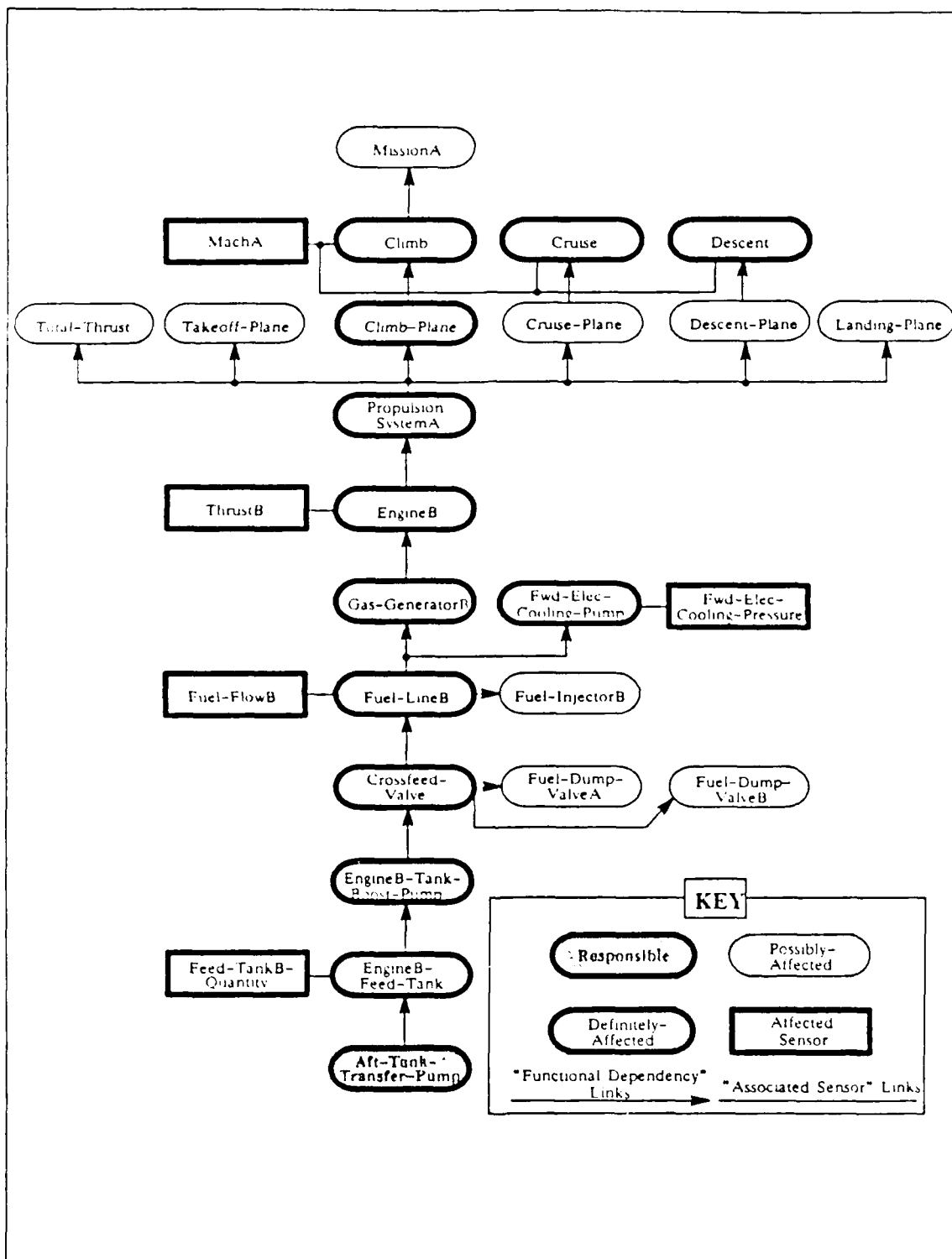


Figure 46 Test 2D Fault Tree Diagram (third hypothesis)

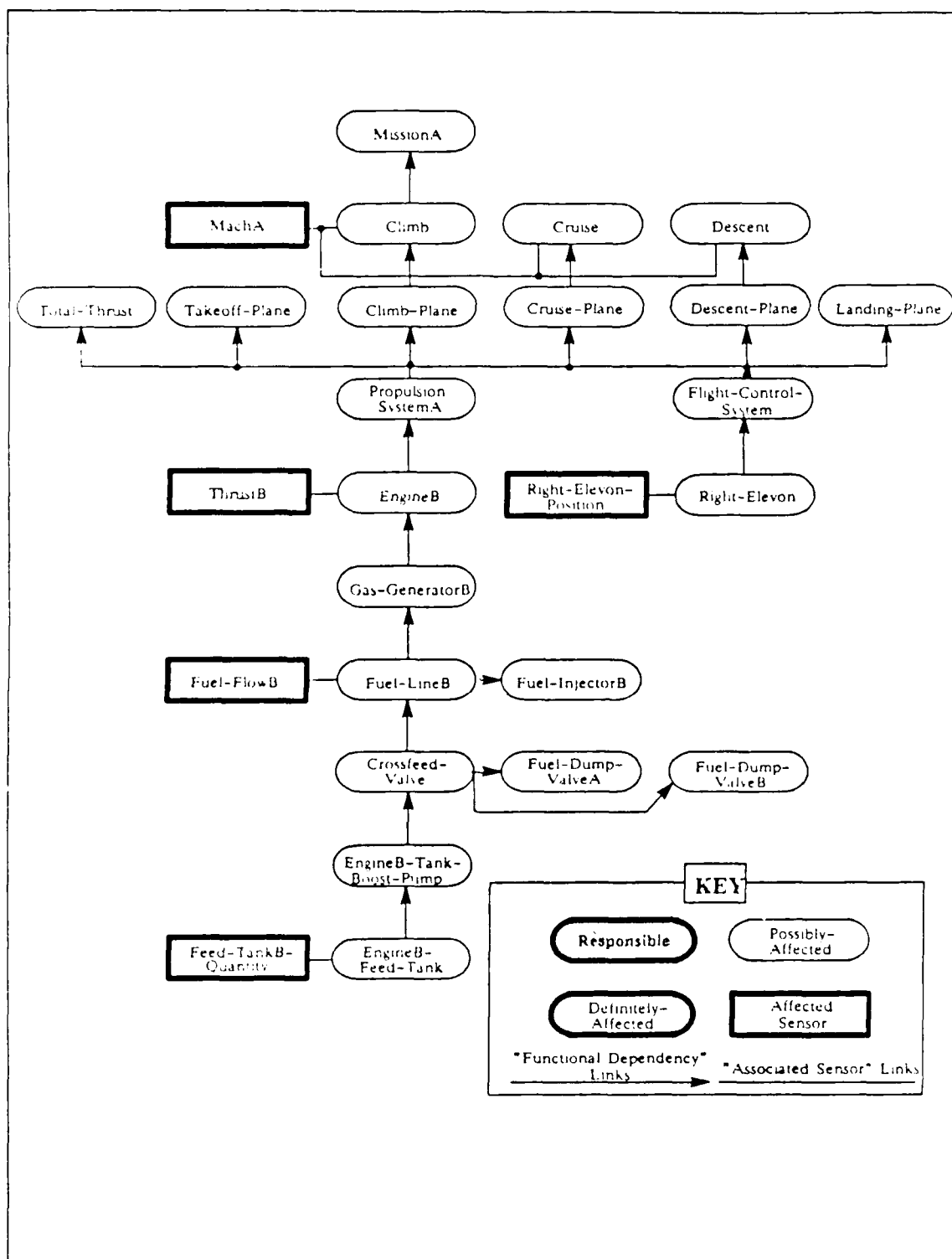


Figure 47. Test 2E Fault Tree Diagram.

VI. Conclusions and Recommendations

6.1 Conclusions

Based on the theoretical development and implementation of the prototype National Aerospace Plane System Status Monitor, the following general conclusions are drawn:

1. It is useful to represent the diagnostic process as a hierarchy of functions, each operating upon and building upon the output of the lower levels of the hierarchy.
2. A combination of traditional expert system techniques and deeper functional reasoning can lead to a more flexible diagnosis system than would be expected if either technique were used alone.
3. A systematic hierarchical representation of a physical system and its functions can aid in both acquiring system knowledge and in the development of an effective diagnostic process.
4. The hierarchical functional representation of the NASP allows the SSM to both diagnose the causes of fault symptoms and determine their effect on all functional levels of the aircraft and its mission.

One specific conclusion can also be drawn from this study of a prototype National Aerospace Plane System Status Monitor.

1. The absence of one key symptom, or the addition of one extraneous symptom, can prevent both Stages 1 and 2 from producing a valid fault hypothesis. (Also see Specific Recommendation 1.)

6.2 Recommendations

Based on the results of this study and the capabilities of the prototype NASP System Status Monitor, the following general recommendations are made:

1. The SSM should be expanded to implement the numerical modeling capabilities of the monitoring function. This capability would allow the SSM to be operated with a stream of sensor input values to produce an event-driven simulation of a NASP mission.
2. As a project for a future student or group of students, the two highest levels of the diagnosis process, prediction and planning, should be added to the SSM.
3. The SSM displays and other aircrew interfaces should be subjected to a human factors analysis. This analysis would determine the best way to present the SSM information to the aircrew, and how best to receive commands and information from the aircrew.

Based on the details of the prototype NASP System Status Monitor, the following specific recommendations are made:

1. Both Stage 1 and Stage 2 of the diagnosis function should be modified to recognize extraneous symptoms, or the absence of key symptoms. This would allow generation of a valid hypothesis or diagnosis in cases that would otherwise produce default hypotheses.
2. The SSM's knowledge base needs the ability to represent logical relationships. For example, it should be possible to represent and reason about the fact that the hydraulic subsystem pressure is functionally dependent on pump-A and/or pump-B.
3. The capability should be added to interactively enter or automatically infer the first derivative of sensor values. As an example, the user can now specify a symptom such as high or low "Nosecap-Temp." The user should also be able to specify that "Nosecap-Temp" is increasing or decreasing. Each of the diagnosis functions should also be able to reason about these sensor derivative values.
4. Stage 2 of the SSM diagnosis function can build fault hypotheses based only on the primitive components. If no valid hypotheses can be produced starting with the primitive components, Stage 2 should be able to move up one level in the fault tree.

AD-A190 681

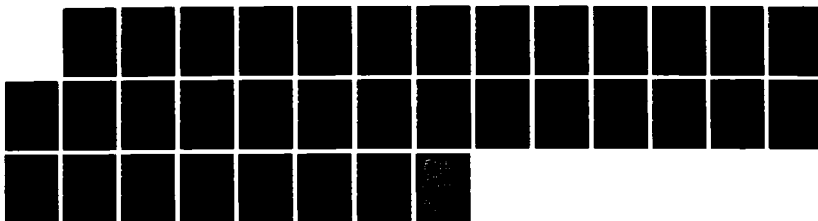
A SYSTEM STATUS MONITOR FOR THE NATIONAL AEROSPACE
PLANE(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
SCHOOL OF ENGINEERING J M BRAUNHANN DEC 87

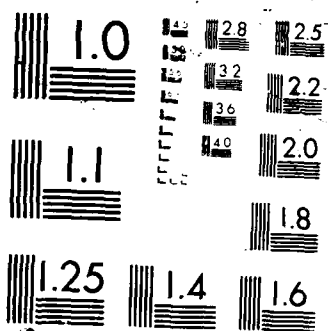
2/2

UNCLASSIFIED

AFIT/SCE/ENG/87D-1

F/G 1/3. 12 NL





the diagnosis process over again. This would ensure that higher-level fault symptoms were diagnosed at least to the subsystem or system level.

5. Currently, the SSM shows only engine instruments in the upper right portion of the display. The interface functions should be expanded to show instrument displays appropriate to the pictorial display in the upper left portion of the SSM display. An example would be to display fuel gauges when the fuel system is pictured in the upper left display window.

Appendix A. *Explanation of the NASP Aircraft Systems*

The current representation of the National Aerospace Plane in the NASP System Status Monitor knowledge base contains five aircraft systems. These systems are

1. Propulsion system
2. Hydraulic system
3. Fuel system
4. Flight controls system
5. Thermal protection system.

Following is an explanation of the physical properties of the five aircraft systems represented in the knowledge base.

A.1 Propulsion System

The propulsion system consists of two engines. The engines are modeled after the airturbo ramjet (ATR) as described in [21]. This engine uses a gas generator supplied with cryogenic fuel such as liquid hydrogen/liquid oxygen. The fuel combines in the gas generator and expands through the turbine to power the compressor. Unburned fuel is combined in the combustor with compressed air from the compressor. The compressor is only needed at low Mach numbers (less than Mach 2-3). At higher Mach numbers, ram-air is sufficient to support combustion in the combustor. Extra hydrogen fuel is added in the combustor by the fuel injectors. The hot fuel exhaust is expanded out the nozzle to produce thrust.

Through the gearbox, each engine drives a hydraulic pump and an electric generator. The "N1" sensor measures the rotational speed of the compressor. The "EGT" sensor measures the gas temperature at the inlet to the turbine. The "EPR" sensor measures the pressure ratio between the compressor inlet and the turbine outlet. The remaining sensors are self-explanatory.

A.2 Hydraulic System

The hydraulic system consists of two identical and independent subsystems. Each hydraulic subsystem consists of a hydraulic fluid reservoir, an engine-driven and electrically-driven pump, and an output line. The reservoir is instrumented with a quantity sensor. Each pump has a pressure sensor, as does the output line.

A.3 Fuel System

The fuel is stored in two primary fuel tanks (forward and aft). Pumps in each of these tanks transfer fuel into left and right feed tanks, where boost pumps move the fuel to the crossfeed valve. The crossfeed valve directs the fuel into the left and/or right fuel lines, from which the cryogenic fuel is fed to the engines and the thermal protection system. Each fuel line has a fuel dump valve. Each of the four fuel tanks has a fuel quantity sensor, and the fuel lines are instrumented with fuel flow sensors.

A.4 Flight Control System

The flight control system consists of four primary control surfaces; right and left elevons, body flap, and rudder. Each control surface is driven by two control surface actuators, and is instrumented by a position sensor.

A.5 Thermal Protection System

The thermal protection system works by circulating cryogenic fuel through the hot structures of the aircraft. These hot structures include the leading edges of the nosecap, wings, and vertical tail, and the inlet, nozzle, and internal structure of the engines. Each of the hot structures has an associated temperature sensor. The cryogenic fuel is forced through the thermal protection system by six pumps, each of which has a pressure sensor.

Appendix B. Listing of the NASP System Status Monitor Knowledge Base

```
(
(MissionA mission (parts (Takeoff
                          Climb
                          Cruise
                          Descent
                          Landing))
  (functionally-dependent-on (Takeoff
                              Climb
                              Cruise
                              Descent
                              Landing)))

(Takeoff flight-phase (part-of (MissionA))
  (parts (Takeoff-Plane
          Total-Thrust
          Weight
          Drag
          Attitude
          Lift
          AirspeedA
          AltitudeA
          Climb-RateA))
  (associated-sensors (AirspeedA
                      AltitudeA
                      Climb-RateA))
  (functional-dependents (MissionA
                          Climb
                          Cruise
                          Descent
                          Landing))
  (functionally-dependent-on (Takeoff-Plane
                              Total-Thrust
                              Weight
                              Drag
                              Attitude
                              Lift)))

(Climb flight-phase (part-of (MissionA))
  (parts (Climb-Plane
          Total-Thrust
          Weight
          Drag
          Attitude
          Lift
          MachA
          AltitudeA
          Climb-RateA))
  (associated-sensors (MachA
                      AltitudeA
                      Climb-RateA))
  (functional-dependents (MissionA
                          Cruise
                          Descent
                          Landing))
  (functionally-dependent-on (Takeoff
                              Climb-Plane
                              Total-Thrust
                              Weight
                              Drag
                              Attitude
                              Lift)))

(Cruise flight-phase (part-of (MissionA))
  (parts (Cruise-Plane
          Total-Thrust
          Weight
          Drag
          Attitude
```

```

        Lift
        MachA
        AltitudeA))
(associated-sensors (MachA
                    AltitudeA))
(functional-dependents (MissionA
                      Descent
                      Landing))
(functionally-dependent-on (Takeoff
                          Climb
                          Cruise-Plane
                          Total-Thrust
                          Weight
                          Drag
                          Attitude
                          Lift)))

(Descent flight-phase (part-of (MissionA))
  (parts (Descent-Plane
        Weight
        Drag
        Attitude
        Lift
        MachA
        AltitudeA
        Sink-RateA))
    (associated-sensors (MachA
                        AltitudeA
                        Sink-RateA))
    (functional-dependents (MissionA
                          Landing))
    (functionally-dependent-on (Takeoff
                              Climb
                              Cruise
                              Descent-Plane
                              Weight
                              Drag
                              Attitude
                              Lift))))

(Landing flight-phase (part-of (MissionA))
  (parts (Landing-Plane
        Total-Thrust
        Weight
        Drag
        Attitude
        Lift
        AirspeedA
        AltitudeA
        Sink-RateA))
    (associated-sensors (AirspeedA
                        AltitudeA
                        Sink-RateA))
    (functional-dependents (MissionA))
    (functionally-dependent-on (Takeoff
                              Climb
                              Cruise
                              Descent
                              Landing-Plane
                              Weight
                              Drag
                              Attitude
                              Lift))))

(Total-Thrust flight-parameter (part-of (Takeoff
        Climb
        Cruise
        Landing))
  (parts (ThrustA
        ThrustB))
  (associated-sensors (ThrustA
                      ThrustB)))

```

```

      (functional-dependents (Takeoff
                             Climb
                             Cruise
                             Landing))
      (functionally-dependent-on (Propulsion-SystemA
                                AirspeedA
                                MachA
                                AltitudeA))
      (causes (((increase ThrustA)(increase Total-Thrust))
                ((decrease ThrustA)(decrease Total-Thrust))
                ((increase ThrustB)(increase Total-Thrust))
                ((decrease ThrustB)(decrease Total-Thrust)))))

(Weight flight-parameter (part-of (Takeoff
                                   Climb
                                   Cruise
                                   Descent
                                   Landing))
  (associated-sensors (Total-Fuel-Quantity))
  (functional-dependents (Takeoff
                          Climb
                          Cruise
                          Descent
                          Landing))
  (functionally-dependent-on (Fwd-Tank-Quantity
                              Aft-Tank-Quantity
                              Feed-TankA-Quantity
                              Feed-TankB-Quantity))
  (causes (((decrease Total-Fuel-Quantity)(decrease Weight)))))

(Drag flight-parameter (part-of (Takeoff
                                 Climb
                                 Cruise
                                 Descent
                                 Landing))
  (functional-dependents (Takeoff
                          Climb
                          Cruise
                          Descent
                          Landing))
  (functionally-dependent-on (Lift
                              AirspeedA
                              MachA
                              Attitude
                              Left-Elevon-Position
                              Right-Elevon-Position
                              Body-Flap-Position
                              Rudder-Position))
  (causes (((decrease Weight)(decrease Drag))
            ((increase Left-Elevon-Position)(increase Drag))
            ((decrease Left-Elevon-Position)(decrease Drag))
            ((increase Right-Elevon-Position)(increase Drag))
            ((decrease Right-Elevon-Position)(decrease Drag))
            ((increase Body-Flap-Position)(increase Drag))
            ((decrease Body-Flap-Position)(decrease Drag))
            ((increase Rudder-Position)(increase Drag))
            ((decrease Rudder-Position)(decrease Drag)))))

(Attitude flight-parameter (part-of (Takeoff
                                     Climb
                                     Cruise
                                     Descent
                                     Landing))
  (parts (PitchA
          RollA
          YawA))
  (associated-sensors (PitchA
                      RollA
                      YawA))
  (functional-dependents (Takeoff
                          Climb
                          Cruise
                          Descent

```

```

                                Landing
                                Drag
                                Lift))
    (functionally-dependent-on (Left-Elevon-Position
                                Right-Elevon-Position
                                Body-Flap-Position
                                Rudder-Position)))

(Lift flight-parameter (part-of (Takeoff
                                Climb
                                Cruise
                                Descent
                                Landing))
    (functional-dependents (Takeoff
                            Climb
                            Cruise
                            Descent
                            Landing
                            Drag))
    (functionally-dependent-on (Weight
                                Attitude
                                AirspeedA
                                MachA
                                Left-Elevon-Position
                                Right-Elevon-Position
                                Body-Flap-Position)))

(AirspeedA aircraft-sensor (part-of (Takeoff
                                Landing))
    (associated-component (Takeoff
                            Landing))
    (association-type ((Takeoff parameter)
                        (Landing parameter)))
    (functional-dependents (Total-Thrust
                            Drag
                            Lift))
    (causes (((increase Total-Thrust)(increase AirspeedA))
              ((decrease Total-Thrust)(decrease AirspeedA))
              ((increase Drag)(decrease AirspeedA))
              ((decrease Drag)(increase AirspeedA))
              ((increase PitchA)(decrease AirspeedA))
              ((decrease PitchA)(increase AirspeedA))
              ((increase AltitudeA)(decrease AirspeedA))
              ((decrease AltitudeA)(increase AirspeedA)))))

(AltitudeA aircraft-sensor (part-of (Takeoff
                                Climb
                                Cruise
                                Descent
                                Landing))
    (associated-component (Takeoff
                            Climb
                            Cruise
                            Descent
                            Landing))
    (association-type ((Takeoff parameter)
                        (Climb parameter)
                        (Cruise parameter)
                        (Descent parameter)
                        (Landing parameter)))
    (functional-dependents (Total-Thrust))
    (causes (((increase AirspeedA)(increase AltitudeA))
              ((decrease AirspeedA)(decrease AltitudeA))
              ((increase MachA)(increase AltitudeA))
              ((decrease MachA)(decrease AltitudeA))
              ((increase PitchA)(increase AltitudeA))
              ((decrease PitchA)(decrease AltitudeA)))))

(Climb-RateA aircraft-sensor (part-of (Takeoff
                                Climb))
    (associated-component (Takeoff
                            Climb))
    (association-type ((Takeoff parameter)

```

```

                                (Climb parameter)))
    (causes (((increase AirspeedA)(increase Climb-RateA))
              ((decrease AirspeedA)(decrease Climb-RateA))
              ((increase MachA)(increase Climb-RateA))
              ((decrease MachA)(decrease Climb-RateA))
              ((increase PitchA)(increase Climb-RateA))
              ((decrease PitchA)(decrease Climb-RateA))
              ((decrease Weight)(increase Climb-RateA)))))

(MachA aircraft-sensor (part-of (Climb
                                Cruise
                                Descent))
  (associated-component (Climb
                        Cruise
                        Descent))
  (association-type ((Climb parameter)
                    (Cruise parameter)
                    (Descent parameter)))
  (functional-dependents (Total-Thrust
                        Drag
                        Lift))
  (causes (((increase Total-Thrust)(increase MachA))
            ((decrease Total-Thrust)(decrease MachA))
            ((decrease Drag)(increase MachA))
            ((increase Drag)(decrease MachA))
            ((decrease PitchA)(increase MachA))
            ((increase PitchA)(decrease MachA))
            ((decrease AltitudeA)(increase MachA))
            ((increase AltitudeA)(decrease MachA)))))

(Sink-RateA aircraft-sensor (part-of (Descent
                                      Landing))
  (associated-component (Descent
                        Landing))
  (association-type ((Descent parameter)
                    (Landing parameter)))
  (causes (((increase AirspeedA)(increase Sink-RateA))
            ((decrease AirspeedA)(decrease Sink-RateA))
            ((increase MachA)(increase Sink-RateA))
            ((decrease MachA)(decrease Sink-RateA))
            ((decrease PitchA)(increase Sink-RateA))
            ((increase PitchA)(decrease Sink-RateA))
            ((decrease Weight)(decrease Sink-RateA)))))

(PitchA aircraft-sensor (part-of (Attitude))
  (associated-component (Attitude))
  (association-type ((Attitude parameter)))
  (causes (((increase Body-Flap-Position)(increase PitchA))
            ((decrease Body-Flap-Position)(decrease PitchA))
            ((increase Fuel-Imbalance)(increase PitchA))
            ((decrease Fuel-Imbalance)(decrease PitchA)))))

(RollA aircraft-sensor (part-of (Attitude))
  (associated-component (Attitude))
  (association-type ((Attitude parameter)))
  (causes (((decrease Left-Elevon-Position)(increase RollA))
            ((increase Left-Elevon-Position)(decrease RollA))
            ((increase Right-Elevon-Position)(increase RollA))
            ((decrease Right-Elevon-Position)(decrease RollA)))))

(YawA aircraft-sensor (part-of (Attitude))
  (associated-component (Attitude))
  (association-type ((Attitude parameter)))
  (causes (((increase Rudder-Position)(increase YawA))
            ((decrease Rudder-Position)(decrease YawA)))))

(Takeoff-Plane plane (part-of (Takeoff))
  (parts (Propulsion-SystemA
          Hydraulic-SystemA
          Fuel-SystemA
          Flight-Control-SysA))
  (functional-dependents (Takeoff))
  (functionally-dependent-on (Propulsion-SystemA
                             Hydraulic-SystemA
                             Fuel-SystemA
                             Flight-Control-SysA)))

```

```

Hydraulic-SystemA
Fuel-SystemA
Flight-Control-SysA)))

(Climb-Plane plane (part-of (Climb))
  (parts (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA
    Thermal-Protection-SysA))
  (functional-dependents (Climb))
  (functionally-dependent-on (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA
    Thermal-Protection-SysA)))

(Cruise-Plane plane (part-of (Cruise))
  (parts (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA
    Thermal-Protection-SysA))
  (functional-dependents (Cruise))
  (functionally-dependent-on (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA
    Thermal-Protection-SysA)))

(Descent-Plane plane (part-of (Descent))
  (parts (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA
    Thermal-Protection-SysA))
  (functional-dependents (Descent))
  (functionally-dependent-on (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA
    Thermal-Protection-SysA)))

(Landing-Plane plane (part-of (Landing))
  (parts (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA))
  (functional-dependents (Landing))
  (functionally-dependent-on (Propulsion-SystemA
    Hydraulic-SystemA
    Fuel-SystemA
    Flight-Control-SysA)))

(Propulsion-SystemA aircraft-system (part-of (Takeoff-Plane
  Climb-Plane
  Cruise-Plane
  Descent-Plane
  Landing-Plane))
  (parts (EngineA
    EngineB))
  (functional-dependents (Takeoff-Plane
    Climb-Plane
    Cruise-Plane
    Descent-Plane
    Landing-Plane
    Total-Thrust))
  (functionally-dependent-on (EngineA
    EngineB)))

(EngineA engine (part-of (Propulsion-SystemA))

```

```

(parts (InletA
        CompressorA
        GearboxA
        Electric-GeneratorA
        Gas-GeneratorA
        TurbineA
        Fuel-InjectorA
        CombustorA
        NozzleA
        N1A
        EprA
        EgtA
        VoltageA
        ThrustA
        VibrationA))
(associated-sensors (VibrationA
                    ThrustA))
(functional-dependents (Propulsion-SystemA
                    ThrustA
                    VibrationA))
(functionally-dependent-on (InletA
                            CompressorA
                            GearboxA
                            Gas-GeneratorA
                            TurbineA
                            Fuel-InjectorA
                            CombustorA
                            NozzleA)))

(InletA engine-component (part-of (EngineA))
                        (associated-sensors (Engine-Inlet-Temp))
                        (functional-dependents (EngineA
                                                CompressorA)))

(CompressorA engine-component (part-of (EngineA))
                             (associated-sensors (N1A
                                                    EprA))
                             (functional-dependents (EngineA
                                                    N1A
                                                    EprA
                                                    CombustorA))
                             (physical-dependents (N1A
                                                    GearboxA))
                             (functionally-dependent-on (GearboxA)))

(GearboxA engine-component (part-of (EngineA))
                          (functional-dependents (EngineA
                                                    CompressorA
                                                    Electric-GeneratorA
                                                    Engine-Hyd-PumpA1))
                          (functionally-dependent-on (TurbineA)))

(Electric-GeneratorA engine-component (part-of (EngineA))
                                     (associated-sensors (VoltageA))
                                     (functional-dependents (Electric-Hyd-PumpA2
                                                            Fwd-Elec-Cooling-Pump
                                                            Left-Elec-Cooling-Pump))
                                     (functionally-dependent-on (GearboxA)))

(Gas-GeneratorA engine-component (part-of (EngineA))
                                (functional-dependents (EngineA
                                                        TurbineA))
                                (functionally-dependent-on (Fuel-LineA))
                                (physical-dependents (TurbineA)))

(TurbineA engine-component (part-of (EngineA))
                          (associated-sensors (EgtA))
                          (functional-dependents (EngineA
                                                  EgtA
                                                  GearboxA))
                          (physical-dependents (EgtA
                                                  Fuel-LineA)))

```

```

(Fuel-InjectorA engine-component (part-of (EngineA))
  (functional-dependents (EngineA
    CombustorA))
  (functionally-dependent-on (Fuel-InjectorA)))

(CombustorA engine-component (part-of (EngineA))
  (associated-sensors (EprA))
  (functional-dependents (EngineA
    EprA))
  (functionally-dependent-on (CompressorA
    Fuel-InjectorA)))

(NozzleA engine-component (part-of (EngineA))
  (functional-dependents (EngineA
    ThrustA)))

(N1A engine-sensor (part-of (EngineA))
  (associated-component (CompressorA))
  (association-type ((CompressorA parameter)))
  (causes (((increase EprA)(increase N1A))
    ((decrease EprA)(decrease N1A)))))

(EprA engine-sensor (part-of (EngineA))
  (associated-component (CompressorA
    CombustorA))
  (association-type ((CombustorA output)
    (CompressorA input)))
  (causes (((increase N1A)(increase EprA))
    ((decrease N1A)(decrease EprA)))))

(EgtA engine-sensor (part-of (EngineA))
  (associated-component (TurbineA))
  (association-type ((TurbineA output)))
  (causes (((increase N1A)(decrease EgtA))
    ((decrease N1A)(increase EgtA))
    ((increase Fuel-FlowA)(increase EgtA))
    ((decrease Fuel-FlowA)(decrease EgtA)))))

(VoltageA engine-sensor (part-of (EngineA))
  (associated-component (Electric-GeneratorA))
  (association-type ((Electric-GeneratorA output)))
  (causes (((increase N1A)(increase VoltageA))
    ((decrease N1A)(decrease VoltageA)))))

(ThrustA engine-sensor (part-of (EngineA))
  (associated-component (EngineA
    Total-Thrust))
  (association-type ((EngineA output)
    (Total-Thrust parameter)))
  (causes (((increase EprA)(increase ThrustA))
    ((decrease EprA)(decrease ThrustA))
    ((increase Fuel-FlowA)(increase ThrustA))
    ((decrease Fuel-FlowA)(decrease ThrustA)))))

(VibrationA engine-sensor (part-of (EngineA))
  (associated-component (EngineA))
  (association-type ((EngineA parameter)))
  (causes (((increase N1A)(increase VibrationA))
    ((decrease N1A)(decrease VibrationA)))))

(EngineB engine (part-of (Propulsion-SystemA))
  (parts (InletB
    CompressorB
    GearboxB
    Electric-GeneratorB
    Gas-GeneratorB
    TurbineB
    Fuel-InjectorB
    CombustorB
    NozzleB
    N1B
    EprB

```

```

        EgtB
        VoltageB
        ThrustB
        VibrationB))
    (associated-sensors (VibrationB
                        ThrustB))
    (functionally-dependent-on (InletB
                                CompressorB
                                GearboxB
                                Gas-GeneratorB
                                TurbineB
                                Fuel-InjectorB
                                CombustorB
                                NozzleB))
    (functional-dependents (Propulsion-SystemA
                            ThrustB
                            VibrationB)))

(InletB engine-component (part-of (EngineB))
  (associated-sensors (Engine-Inlet-Temp))
  (functional-dependents (EngineB
                          CompressorB)))

(CompressorB engine-component (part-of (EngineB))
  (associated-sensors (N1B
                      EprB))
  (functional-dependents (EngineB
                          N1B
                          EprB
                          CombustorB))
  (physical-dependents (N1B
                        GearboxB))
  (functionally-dependent-on (GearboxB)))

(GearboxB engine-component (part-of (EngineB))
  (functional-dependents (EngineB
                          CompressorB
                          Electric-GeneratorB
                          Engine-Hyd-PumpB1))
  (functionally-dependent-on (TurbineB)))

(Electric-GeneratorB engine-component (part-of (EngineB))
  (associated-sensors (VoltageB))
  (functional-dependents (Electric-Hyd-PumpB2
                          Right-Elec-Cooling-Pump))
  (functionally-dependent-on (GearboxB)))

(Gas-GeneratorB engine-component (part-of (EngineB))
  (functional-dependents (EngineB
                          TurbineB))
  (functionally-dependent-on (Fuel-LineB))
  (physical-dependents (TurbineB)))

(TurbineB engine-component (part-of (EngineB))
  (associated-sensors (EgtB))
  (functional-dependents (EngineB
                          EgtB
                          GearboxB))
  (physical-dependents (EgtB)))

(Fuel-InjectorB engine-component (part-of (EngineB))
  (functional-dependents (EngineB
                          CombustorB))
  (functionally-dependent-on (Fuel-LineB)))

(CombustorB engine-component (part-of (EngineB))
  (associated-sensors (EprB))
  (functional-dependents (EngineB
                          EprB))
  (functionally-dependent-on (CompressorB
                              Fuel-InjectorB)))

(NozzleB engine-component (part-of (EngineB)))

```

```

      (functional-dependents (EngineB
                             ThrustB)))

(N1B engine-sensor (part-of (EngineB))
  (associated-component (CompressorB))
  (association-type ((CompressorB parameter)))
  (causes (((increase EprB)(increase N1B))
            ((decrease EprB)(decrease N1B)))))

(EprB engine-sensor (part-of (EngineB))
  (associated-component (CompressorB
                        CombustorB))
  (association-type ((CombustorB output)
                    (CompressorB input)))
  (causes (((increase N1B)(increase EprB))
            ((decrease N1B)(decrease EprB)))))

(EgtB engine-sensor (part-of (EngineB))
  (associated-component (TurbineB))
  (association-type ((TurbineB output)))
  (causes (((increase N1B)(decrease EgtB))
            ((decrease N1B)(increase EgtB))
            ((increase Fuel-FlowB)(increase EgtB))
            ((decrease Fuel-FlowB)(decrease EgtB)))))

(VoltageB engine-sensor (part-of (EngineB))
  (associated-component (Electric-GeneratorB))
  (association-type ((Electric-GeneratorB output)))
  (causes (((increase N1B)(increase VoltageB))
            ((decrease N1B)(decrease VoltageB)))))

(ThrustB engine-sensor (part-of (EngineB))
  (associated-component (EngineB
                        Total-Thrust))
  (association-type ((EngineB output)
                    (Total-Thrust)))
  (causes (((increase EprB)(increase ThrustB))
            ((decrease EprB)(decrease ThrustB))
            ((increase Fuel-FlowB)(increase ThrustB))
            ((decrease Fuel-FlowB)(decrease ThrustB)))))

(VibrationB engine-sensor (part-of (EngineB))
  (associated-component (EngineB))
  (association-type ((EngineB parameter)))
  (causes (((increase N1B)(increase VibrationB))
            ((decrease N1B)(decrease VibrationB)))))

(Hydraulic-SystemA aircraft-system (part-of (Takeoff-Plane
                                              Climb-Plane
                                              Cruise-Plane
                                              Descent-Plane
                                              Landing-Plane))
  (parts (Hydraulic-SubsystemA
          Hydraulic-SubsystemB))
  (functional-dependents (Takeoff-Plane
                          Climb-Plane
                          Cruise-Plane
                          Descent-Plane
                          Landing-Plane))
  (functionally-dependent-on (Hydraulic-SubsystemA
                              Hydraulic-SubsystemB)))

(Hydraulic-SubsystemA hydraulic-subsystem (part-of (Hydraulic-SystemA))
  (parts (Hydraulic-LineA
          Hyd-SubsysA-Pressure
          Engine-Hyd-PumpA1
          Electric-Hyd-PumpA2
          Hydraulic-ReservoirA
          Hyd-PumpA1-Pressure
          Hyd-PumpA2-Pressure

```

```

Hyd-QuantityA))
(functional-dependents (Hydraulic-SystemA))
(functionally-dependent-on (Hydraulic-LineA)))

(Hydraulic-LineA hydraulic-line (part-of (Hydraulic-SubsystemA))
  (associated-sensors (Hyd-SubsysA-Pressure))
  (functional-dependents (Hydraulic-SubsystemA
    Hyd-SubsysA-Pressure
    Left-Elevon-Actuator-1
    Right-Elevon-Actuator-1
    Body-Flap-Actuator-1
    Rudder-Actuator-1
    Right-Hyd-Cooling-Pump))
  (functionally-dependent-on (Engine-Hyd-PumpA1
    Electric-Hyd-PumpA2))
  (physical-dependents (Hyd-SubsysA-Pressure)))

(Hyd-SubsysA-Pressure hyd-pressure (part-of (Hydraulic-SubsystemA))
  (associated-component (Hydraulic-LineA))
  (association-type ((Hydraulic-LineA parameter)))))

(Engine-Hyd-PumpA1 hydraulic-pump (part-of (Hydraulic-SubsystemA))
  (associated-sensors (Hyd-PumpA1-Pressure))
  (functionally-dependent-on (GearboxA
    Hydraulic-ReservoirA))
  (functional-dependents (Hydraulic-LineA)))

(Electric-Hyd-PumpA2 hydraulic-pump (part-of (Hydraulic-SubsystemA))
  (associated-sensors (Hyd-PumpA2-Pressure))
  (functionally-dependent-on (Electric-GeneratorA
    Hydraulic-ReservoirA))
  (functional-dependents (Hydraulic-LineA)))

(Hydraulic-ReservoirA hydraulic-reservoir (part-of (Hydraulic-SubsystemA))
  (associated-sensors (Hyd-QuantityA))
  (functional-dependents (Engine-Hyd-PumpA1
    Electric-Hyd-PumpA2)))

(Hyd-PumpA1-Pressure hyd-pressure (part-of (Hydraulic-SubsystemA))
  (associated-component (Engine-Hyd-PumpA1))
  (association-type ((Engine-Hyd-PumpA1 parameter)
)))

(Hyd-PumpA2-Pressure hyd-pressure (part-of (Hydraulic-SubsystemA))
  (associated-component (Electric-Hyd-PumpA2))
  (association-type ((Electric-Hyd-PumpA2 parameter)))))

(Hyd-QuantityA hyd-quantity (part-of (Hydraulic-SubsystemA))
  (associated-component (Hydraulic-ReservoirA))
  (association-type ((Hydraulic-ReservoirA parameter)))))

(Hydraulic-SubsystemB hydraulic-subsystem (part-of (Hydraulic-SystemA))
  (parts (Hydraulic-LineB
    Hyd-SubsysB-Pressure
    Engine-Hyd-PumpB1
    Electric-Hyd-PumpB2
    Hydraulic-ReservoirB
    Hyd-PumpB1-Pressure
    Hyd-PumpB2-Pressure
    Hyd-QuantityB))
  (functional-dependents (Hydraulic-SystemA))
  (functionally-dependent-on (Hydraulic-LineB)))

(Hydraulic-LineB hydraulic-line (part-of (Hydraulic-SubsystemB))
  (associated-sensors (Hyd-SubsysB-Pressure))
  (functional-dependents (Hydraulic-SubsystemB
    Hyd-SubsysB-Pressure
    Left-Elevon-Actuator-2
    Right-Elevon-Actuator-2
    Body-Flap-Actuator-2
    Rudder-Actuator-2

```

```

                                Fwd-Hyd-Cooling-Pump
                                Left-Hyd-Cooling-Pump))
    (physical-dependents (Hyd-SubsysB-Pressure)))

(Hyd-SubsysB-Pressure hyd-pressure (part-of (Hydraulic-SubsystemB))
  (associated-component (Hydraulic-LineB))
  (association-type ((Hydraulic-LineB parameter))))

(Engine-Hyd-PumpB1 hydraulic-pump (part-of (Hydraulic-SubsystemB))
  (associated-sensors (Hyd-PumpB1-Pressure))
  (functionally-dependent-on (GearboxB
                              Hydraulic-ReservoirB))
  (functional-dependents (Hydraulic-LineB)))

(Electric-Hyd-PumpB2 hydraulic-pump (part-of (Hydraulic-SubsystemB))
  (associated-sensors (Hyd-PumpB2-Pressure))
  (functionally-dependent-on (Electric-GeneratorB
                              Hydraulic-ReservoirB))
  (functional-dependents (Hydraulic-LineB)))

(Hydraulic-ReservoirB hydraulic-reservoir (part-of (Hydraulic-SubsystemB))
  (associated-sensors (Hyd-QuantityB))
  (functional-dependents (Engine-Hyd-PumpB1
                          Electric-Hyd-PumpB2)))

(Hyd-PumpB1-Pressure hyd-pressure (part-of (Hydraulic-SubsystemB))
  (associated-component (Engine-Hyd-PumpB1))
  (association-type ((Engine-Hyd-PumpB1 parameter))))

(Hyd-PumpB2-Pressure hyd-pressure (part-of (Hydraulic-SubsystemB))
  (associated-component (Electric-Hyd-PumpB2))
  (association-type ((Electric-Hyd-PumpB2 parameter))))

(Hyd-QuantityB hyd-quantity (part-of (Hydraulic-SubsystemB))
  (associated-component (Hydraulic-ReservoirB))
  (association-type ((Hydraulic-ReservoirB parameter))))

(Fuel-SystemA aircraft-system (part-of (Takeoff-Plane
                                         Climb-Plane
                                         Cruise-Plane
                                         Descent-Plane
                                         Landing-Plane))
  (parts (Fwd-Fuel-Tank
          Aft-Fuel-Tank
          EngineA-Feed-Tank
          EngineB-Feed-Tank
          Fwd-Tank-Transfer-Pump
          Aft-Tank-Transfer-Pump
          EngineA-Tank-Boost-Pump
          EngineB-Tank-Boost-Pump
          Fuel-LineA
          Fuel-LineB
          Crossfeed-Valve
          Fuel-Dump-ValveA
          Fuel-Dump-ValveB
          Fuel-FlowA
          Fuel-FlowB
          Fwd-Tank-Quantity
          Aft-Tank-Quantity
          Feed-TankA-Quantity
          Feed-TankB-Quantity
          Total-Fuel-Quantity
          Boost-PumpA-Pressure
          Boost-PumpB-Pressure
          Fuel-Imbalance))
  (associated-sensors (Total-Fuel-Quantity
                      Fuel-Imbalance))
  (functional-dependents (Takeoff-Plane
                          Climb-Plane
                          Cruise-Plane

```

```

Descent-Plane
Landing-Plane))
(functionally-dependent-on (Fwd-Fuel-Tank
Aft-Fuel-Tank
EngineA-Feed-Tank
EngineB-Feed-Tank
Fuel-LineA
Fuel-LineB
Crossfeed-Valve
Fuel-Dump-ValveA
Fuel-Dump-ValveB)))

(Fwd-Fuel-Tank fuel-tank (part-of (Fuel-SystemA))
  (associated-sensors (Fwd-Tank-Quantity))
  (functional-dependents (Fuel-SystemA
    Fwd-Tank-Transfer-Pump)))

(Aft-Fuel-Tank fuel-tank (part-of (Fuel-SystemA))
  (associated-sensors (Aft-Tank-Quantity))
  (functional-dependents (Fuel-SystemA
    Aft-Tank-Transfer-Pump)))

(EngineA-Feed-Tank fuel-tank (part-of (Fuel-SystemA))
  (associated-sensors (Feed-TankA-Quantity))
  (functional-dependents (Fuel-SystemA
    EngineA-Tank-Boost-Pump)))

(EngineB-Feed-Tank fuel-tank (part-of (Fuel-SystemA))
  (associated-sensors (Feed-TankB-Quantity))
  (functional-dependents (Fuel-SystemA
    EngineB-Tank-Boost-Pump)))

(Fwd-Tank-Transfer-Pump fuel-pump (part-of (Fuel-SystemA))
  (functional-dependents (EngineA-Feed-Tank
    EngineB-Feed-Tank))
  (functionally-dependent-on (Fwd-Fuel-Tank)))

(Aft-Tank-Transfer-Pump fuel-pump (part-of (Fuel-SystemA))
  (functional-dependents (EngineA-Feed-Tank
    EngineB-Feed-Tank))
  (functionally-dependent-on (Aft-Fuel-Tank)))

(EngineA-Tank-Boost-Pump fuel-pump (part-of (Fuel-SystemA))
  (functional-dependents (Crossfeed-Valve))
  (functionally-dependent-on (EngineA-Feed-Tank)))

(EngineB-Tank-Boost-Pump fuel-pump (part-of (Fuel-SystemA))
  (functional-dependents (Crossfeed-Valve))
  (functionally-dependent-on (EngineB-Feed-Tank)))

(Crossfeed-Valve fuel-valve (part-of (Fuel-SystemA))
  (functional-dependents (Fuel-SystemA
    Fuel-Dump-ValveA
    Fuel-Dump-ValveB
    Fuel-LineA
    Fuel-LineB))
  (functionally-dependent-on (EngineA-Tank-Boost-Pump
    EngineB-Tank-Boost-Pump)))

(Fuel-Dump-ValveA fuel-valve (part-of (Fuel-SystemA))
  (functional-dependents (Fuel-SystemA))
  (functionally-dependent-on (Crossfeed-Valve)))

(Fuel-Dump-ValveB fuel-valve (part-of (Fuel-SystemA))
  (functional-dependents (Fuel-SystemA))
  (functionally-dependent-on (Crossfeed-Valve)))

(Fuel-LineA fuel-line (part-of (Fuel-SystemA))
  (associated-sensors (Fuel-FlowA))
  (functional-dependents (Fuel-SystemA
    Gas-GeneratorA
    Fuel-InjectorA
    Fwd-Hyd-Cooling-Pump

```

```

Right-Hyd-Cooling-Pump
Left-Elec-Cooling-Pump)))

(Fuel-LineB fuel-line (part-of (Fuel-SystemA))
  (associated-sensors (Fuel-FlowB))
  (functional-dependents (Fuel-SystemA
    Gas-GeneratorB
    Fuel-InjectorB
    Fwd-Elec-Cooling-Pump
    Right-Elec-Cooling-Pump
    Left-Hyd-Cooling-Pump)))

(Fuel-FlowA fuel-flow (part-of (Fuel-SystemA))
  (associated-component (Fuel-LineA))
  (association-type ((Fuel-LineA parameter)))
  (causes (((increase Boost-PumpA-Pressure)(increase Fuel-FlowA))
    ((decrease Boost-PumpA-Pressure)(decrease Fuel-FlowA)))))

(Fuel-FlowB fuel-flow (part-of (Fuel-SystemA))
  (associated-component (Fuel-LineB))
  (association-type ((Fuel-LineB parameter)))
  (causes (((increase Boost-PumpB-Pressure)(increase Fuel-FlowB))
    ((decrease Boost-PumpB-Pressure)(decrease Fuel-FlowB)))))

(Fwd-Tank-Quantity fuel-qty-sensor (part-of (Fuel-SystemA))
  (associated-component (Fwd-Fuel-Tank))
  (association-type ((Fwd-Fuel-Tank parameter)))
  (functional-dependents (Weight)))

(Aft-Tank-Quantity fuel-qty-sensor (part-of (Fuel-SystemA))
  (associated-component (Aft-Fuel-Tank))
  (association-type ((Aft-Fuel-Tank parameter)))
  (functional-dependents (Weight)))

(Feed-TankA-Quantity fuel-qty-sensor (part-of (Fuel-SystemA))
  (associated-component (EngineA-Feed-Tank))
  (association-type ((EngineA-Feed-Tank parameter)))
  (functional-dependents (Weight)))

(Feed-TankB-Quantity fuel-qty-sensor (part-of (Fuel-SystemA))
  (associated-component (EngineB-Feed-Tank))
  (association-type ((EngineB-Feed-Tank parameter)))
  (functional-dependents (Weight)))

(Boost-PumpA-Pressure sensor (part-of (Fuel-SystemA)))

(Boost-PumpB-Pressure sensor (part-of (Fuel-SystemA)))

(Total-Fuel-Quantity fuel-qty-sensor (part-of (Fuel-SystemA))
  (associated-component (Fuel-SystemA
    Weight))
  (association-type ((Fuel-SystemA parameter)
    (Weight parameter))))

(Fuel-Imbalance fuel-qty-sensor (part-of (Fuel-SystemA))
  (associated-component (Fuel-SystemA))
  (association-type ((Fuel-SystemA parameter))))

(Flight-Control-SysA aircraft-system (part-of (Takeoff-Plane
  Climb-Plane
  Cruise-Plane
  Descent-Plane
  Landing-Plane))
  (parts (Left-Elevon
    Right-Elevon
    Body-Flap
    Rudder
    Control-Stick
    Left-Rudder-Pedal

```

```

Right-Rudder-Pedal
Left-Elevon-Actuator-1
Left-Elevon-Actuator-2
Left-Elevon-Position
Right-Elevon-Actuator-1
Right-Elevon-Actuator-2
Right-Elevon-Position
Body-Flap-Actuator-1
Body-Flap-Actuator-2
Body-Flap-Position
Rudder-Actuator-1
Rudder-Actuator-2
Rudder-Position))
(functional-dependents (Takeoff-Plane
                        Climb-Plane
                        Cruise-Plane
                        Descent-Plane
                        Landing-Plane))
(functionally-dependent-on (Left-Elevon
                            Right-Elevon
                            Body-Flap
                            Rudder)))

(Left-Elevon control-surface (part-of (Flight-Control-SysA))
 (associated-sensors (Left-Elevon-Position))
 (functional-dependents (Flight-Control-SysA
                          Left-Elevon-Position))
 (functionally-dependent-on (Left-Elevon-Actuator-1
                              Left-Elevon-Actuator-2)))

(Right-Elevon control-surface (part-of (Flight-Control-SysA))
 (associated-sensors (Right-Elevon-Position))
 (functional-dependents (Flight-Control-SysA
                          Right-Elevon-Position))
 (functionally-dependent-on (Right-Elevon-Actuator-1
                              Right-Elevon-Actuator-2)))

(Body-Flap control-surface (part-of (Flight-Control-SysA))
 (associated-sensors (Body-Flap-Position))
 (functional-dependents (Flight-Control-SysA
                          Body-Flap-Position))
 (functionally-dependent-on (Body-Flap-Actuator-1
                              Body-Flap-Actuator-2)))

(Rudder control-surface (part-of (Flight-Control-SysA))
 (associated-sensors (Rudder-Position))
 (functional-dependents (Flight-Control-SysA
                          Rudder-Position))
 (functionally-dependent-on (Rudder-Actuator-1
                              Rudder-Actuator-2)))

(Control-Stick component (part-of (Flight-Control-SysA)))

(Left-Rudder-Pedal component (part-of (Flight-Control-SysA)))

(Right-Rudder-Pedal component (part-of (Flight-Control-SysA)))

(Left-Elevon-Actuator-1 control-surface-actuator (part-of (Flight-Control-SysA))
 (functional-dependents (Left-Elevon))
 (functionally-dependent-on (Hydraulic-Line
                              A)))

(Right-Elevon-Actuator-1 control-surface-actuator (part-of (Flight-Control-SysA))
 (functional-dependents (Right-Elevon))
 (functionally-dependent-on (Hydraulic-Line
                              A)))

(Left-Elevon-Actuator-2 control-surface-actuator (part-of (Flight-Control-SysA))
 (functional-dependents (Left-Elevon))
 (functionally-dependent-on (Hydraulic-Line
                              B)))

(Right-Elevon-Actuator-2 control-surface-actuator (part-of (Flight-Control-SysA))

```

```

                                (functional-dependents (Right-Elevon))
                                (functionally-dependent-on (Hydraulic-LineA))
eB)))

(Body-Flap-Actuator-1 control-surface-actuator (part-of (Flight-Control-SysA))
                                                (functional-dependents (Body-Flap))
                                                (functionally-dependent-on (Hydraulic-LineA))
))

(Body-Flap-Actuator-2 control-surface-actuator (part-of (Flight-Control-SysA))
                                                (functional-dependents (Body-Flap))
                                                (functionally-dependent-on (Hydraulic-LineB))
))

(Rudder-Actuator-1 control-surface-actuator (part-of (Flight-Control-SysA))
                                              (functional-dependents (Rudder))
                                              (functionally-dependent-on (Hydraulic-LineA)))

(Rudder-Actuator-2 control-surface-actuator (part-of (Flight-Control-SysA))
                                              (functional-dependents (Rudder))
                                              (functionally-dependent-on (Hydraulic-LineB)))

(Left-Elevon-Position control-surface-position (part-of (Flight-Control-SysA))
                                                (associated-component (Left-Elevon))
                                                (association-type ((Left-Elevon parameter)))
                                                (functional-dependents (Drag
                                                                    Attitude
                                                                    Lift))
                                                (causes (((move-left control-stick)(increase
Left-Elevon-Position))
                                                                    ((move-right control-stick)(decreas
e Left-Elevon-Position)))))

(Right-Elevon-Position control-surface-position (part-of (Flight-Control-SysA))
                                                (associated-component (Right-Elevon))
                                                (association-type ((Right-Elevon parameter))
                                                (functional-dependents (Drag
                                                                    Attitude
                                                                    Lift))
                                                (causes (((move-right control-stick)(increas
e Right-Elevon-Position))
                                                                    ((move-left control-stick)(decreas
e Right-Elevon-Position)))))

(Body-Flap-Position control-surface-position (part-of (Flight-Control-SysA))
                                                (associated-component (Body-Flap))
                                                (association-type ((Body-Flap parameter)))
                                                (functional-dependents (Drag
                                                                    Attitude
                                                                    Lift))
                                                (causes (((move-aft control-stick)(increase Bo
dy-Flap-Position))
                                                                    ((move-forward control-stick)(decreas
e Body-Flap-Position)))))

(Rudder-Position control-surface-position (part-of (Flight-Control-SysA))
                                            (associated-component (Rudder))
                                            (association-type ((Rudder parameter)))
                                            (functional-dependents (Drag
                                                                    Attitude))
                                            (causes (((push Left-Rudder-Pedal)(decrease Rudd
er-Position))
                                                                    ((push Right-Rudder-Pedal)(increase Rudd
er-Position)))))

(Thermal-Protection-SysA aircraft-system (part-of (Climb-Plane
                                                    Cruise-Plane
                                                    Descent-Plane))
                                           (parts (Nosecap-Cooling
                                                    Left-Wing-Cooling

```

```

Right-Wing-Cooling
Engine-Inlet-Cooling
EngineA-Internal-Cooling
EngineB-Internal-Cooling
Engine-Nozzle-Cooling
Vert-Tail-Cooling
Nosecap-Temp
Left-Wing-Temp
Right-Wing-Temp
Engine-inlet-Temp
EngineA-Internal-Temp
EngineB-Internal-Temp
Engine-Nozzle-Temp
Vert-Tail-Temp
Fwd-Hyd-Cooling-Pump
Fwd-Elec-Cooling-Pump
Left-Hyd-Cooling-Pump
Right-Hyd-Cooling-Pump
Left-Elec-Cooling-Pump
Right-Elec-Cooling-Pump
Fwd-Hyd-Cooling-Pressure
Fwd-Elec-Cooling-Pressure
Left-Hyd-Cooling-Pressure
Right-Hyd-Cooling-Pressure
Left-Elec-Cooling-Pressure
Right-Elec-Cooling-Pressure))
(functional-dependents (Climb-Plane
                        Cruise-Plane
                        Descent-Plane))
(functionally-dependent-on (Nosecap-Cooling
                            Left-Wing-Cooling
                            Right-Wing-Cooling
                            Engine-Inlet-Cooling
                            EngineA-Internal-Cooli
ng
ng
                            EngineB-Internal-Cooli
                            Engine-Nozzle-Cooling
                            Vert-Tail-Cooling)))

(Nosecap-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
                                   (associated-sensors (Nosecap-Temp))
                                   (functional-dependents (Thermal-Protection-SysA
                                                           Nosecap-Temp))
                                   (functionally-dependent-on (Fwd-Hyd-Cooling-Pump
                                                             Fwd-Elec-Cooling-Pump)))

(Left-Wing-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
                                   (associated-sensors (Left-Wing-Temp))
                                   (functional-dependents (Thermal-Protection-SysA
                                                           Left-Wing-Temp))
                                   (functionally-dependent-on (Fwd-Hyd-Cooling-Pump
                                                             Fwd-Elec-Cooling-Pump)))

(Right-Wing-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
                                   (associated-sensors (Right-Wing-Temp))
                                   (functional-dependents (Thermal-Protection-SysA
                                                           Right-Wing-Temp))
                                   (functionally-dependent-on (Fwd-Hyd-Cooling-Pump
                                                             Fwd-Elec-Cooling-Pump)))

(Engine-Inlet-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
                                   (associated-sensors (Engine-Inlet-Temp))
                                   (functional-dependents (Thermal-Protection-SysA
                                                           EngineA
                                                           EngineB
                                                           Engine-Inlet-Temp))
                                   (functionally-dependent-on (Left-Hyd-Cooling-Pump
                                                             Right-Hyd-Cooling-Pump
                                                             Left-Elec-Cooling-Pump
                                                             Right-Elec-Cooling-Pump)))

```

```

(EngineA-Internal-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
      (associated-sensors (EngineA-Internal-Temp))
      (functional-dependents (Thermal-Protection-SysA
                              EngineA
                              EngineA-Internal-Temp))
      (functionally-dependent-on (Left-Hyd-Cooling-Pump
                                  Left-Elec-Cooling-Pump)))

(EngineB-Internal-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
      (associated-sensors (EngineB-Internal-Temp))
      (functional-dependents (Thermal-Protection-SysA
                              EngineB
                              EngineB-Internal-Temp))
      (functionally-dependent-on (Right-Hyd-Cooling-Pump
                                  Right-Elec-Cooling-Pump)))

(Engine-Nozzle-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
      (associated-sensors (Engine-Nozzle-Temp))
      (functional-dependents (Thermal-Protection-SysA
                              EngineA
                              EngineB
                              Engine-Nozzle-Cooling))
      (functionally-dependent-on (Left-Hyd-Cooling-Pump
                                  Right-Hyd-Cooling-Pump
                                  Left-Elec-Cooling-Pump
                                  Right-Elec-Cooling-Pump)))

(Vert-Tail-Cooling cooling-subsystem (part-of (Thermal-Protection-SysA))
      (associated-sensors (Vert-Tail-Temp))
      (functional-dependents (Thermal-Protection-SysA
                              Vert-Tail-Temp))
      (functionally-dependent-on (Left-Hyd-Cooling-Pump
                                  Right-Hyd-Cooling-Pump
                                  Left-Elec-Cooling-Pump
                                  Right-Elec-Cooling-Pump)))

(Fwd-Hyd-Cooling-Pump cooling-pump (part-of (Thermal-Protection-SysA))
      (associated-sensors (Fwd-Hyd-Cooling-Pressure))
      (functional-dependents (Nosecap-Cooling
                              Left-Wing-Cooling
                              Right-Wing-Cooling
                              Fwd-Hyd-Cooling-Pressure))
      (functionally-dependent-on (Hydraulic-LineB
                                  Fuel-LineA)))

(Fwd-Elec-Cooling-Pump cooling-pump (part-of (Thermal-Protection-SysA))
      (associated-sensors (Fwd-Elec-Cooling-Pressure))
      (functional-dependents (Nosecap-Cooling
                              Left-Wing-Cooling
                              Right-Wing-Cooling
                              Fwd-Elec-Cooling-Pressure))
      (functionally-dependent-on (Electric-GeneratorA
                                  Fuel-LineB)))

(Left-Hyd-Cooling-Pump cooling-pump (part-of (Thermal-Protection-SysA))
      (associated-sensors (Left-Hyd-Cooling-Pressure))
      (functional-dependents (Engine-Inlet-Cooling
                              EngineA-Internal-Cooling
                              EngineB-Internal-Cooling
                              Engine-Nozzle-Cooling
                              Vert-Tail-Cooling
                              Left-Hyd-Cooling-Pressure))
      (functionally-dependent-on (Hydraulic-LineB
                                  Fuel-LineB)))

(Left-Elec-Cooling-Pump cooling-pump (part-of (Thermal-Protection-SysA))
      (associated-sensors (Left-Elec-Cooling-Pressure))
      (functional-dependents (Engine-Inlet-Cooling
                              EngineA-Internal-Cooling
                              EngineB-Internal-Cooling
                              Engine-Nozzle-Cooling
                              Vert-Tail-Cooling
                              Left-Hyd-Cooling-Pressure)))

```

```

EngineA-Internal-Cooling
EngineB-Internal-Cooling
Engine-Nozzle-Cooling
Vert-Tail-Cooling
Left-Elec-Cooling-Pressure))
(functionally-dependent-on (Electric-GeneratorA
Fuel-LineA)))

(Right-Hyd-Cooling-Pump cooling-pump (part-of (Thermal-Protection-SysA))
(associated-sensors (Right-Hyd-Cooling-Pressure))
(functional-dependents (Engine-Inlet-Cooling
EngineA-Internal-Cooling
EngineB-Internal-Cooling
Engine-Nozzle-Cooling
Vert-Tail-Cooling
Right-Hyd-Cooling-Pressure))
(functionally-dependent-on (Hydraulic-LineA
Fuel-LineA)))

(Right-Elec-Cooling-Pump cooling-pump (part-of (Thermal-Protection-SysA))
(associated-sensors (Right-Elec-Cooling-Pressure))
(functional-dependents (Engine-Inlet-Cooling
EngineA-Internal-Cooling
EngineB-Internal-Cooling
Engine-Nozzle-Cooling
Vert-Tail-Cooling
Right-Elec-Cooling-Pressure))
(functionally-dependent-on (Electric-GeneratorB
Fuel-LineB)))

(Nosecap-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (Nosecap-Cooling))
(association-type ((Nosecap-Cooling parameter)))
(causes (((increase AltitudeA)(decrease Nosecap-Temp))
((decrease AltitudeA)(increase Nosecap-Temp))
((increase AirspeedA)(increase Nosecap-Temp))
((decrease AirspeedA)(decrease Nosecap-Temp))
((increase MachA)(increase Nosecap-Temp))
((decrease MachA)(decrease Nosecap-Temp))
((increase Fwd-Hyd-Cooling-Pressure)(decrease Nosecap-T
mp)))
((decrease Fwd-Hyd-Cooling-Pressure)(increase Nosecap-T
mp)))
((increase Fwd-Elec-Cooling-Pressure)(decrease Nosecap-T
emp)))
((decrease Fwd-Elec-Cooling-Pressure)(increase Nosecap-T
emp))))))

(Left-Wing-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (Left-Wing-Cooling))
(association-type ((Left-Wing-Cooling parameter)))
(causes (((increase AltitudeA)(decrease Left-Wing-Temp))
((decrease AltitudeA)(increase Left-Wing-Temp))
((increase AirspeedA)(increase Left-Wing-Temp))
((decrease AirspeedA)(decrease Left-Wing-Temp))
((increase MachA)(increase Left-Wing-Temp))
((decrease MachA)(decrease Left-Wing-Temp))
((increase Fwd-Hyd-Cooling-Pressure)(decrease Left-Win
g-Temp))
((decrease Fwd-Hyd-Cooling-Pressure)(increase Left-Win
g-Temp))
((increase Fwd-Elec-Cooling-Pressure)(decrease Left-Wi
ng-Temp))
((decrease Fwd-Elec-Cooling-Pressure)(increase Left-Wi
ng-Temp))))))

(Right-Wing-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (Right-Wing-Cooling))
(association-type ((Right-Wing-Cooling parameter)))
(causes (((increase AltitudeA)(decrease Right-Wing-Temp))
((decrease AltitudeA)(increase Right-Wing-Temp))
((increase AirspeedA)(increase Right-Wing-Temp))
((decrease AirspeedA)(decrease Right-Wing-Temp))

```

```

((increase MachA)(increase Right-Wing-Temp))
((decrease MachA)(decrease Right-Wing-Temp))
((increase Fwd-Hyd-Cooling-Pressure)(decrease Right-W
ing-Temp))
((decrease Fwd-Hyd-Cooling-Pressure)(increase Right-W
ing-Temp))
((increase Fwd-Elec-Cooling-Pressure)(decrease Right-
Wing-Temp))
((decrease Fwd-Elec-Cooling-Pressure)(increase Right-
Wing-Temp))))

(Engine-Inlet-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (Engine-Inlet-Cooling))
(association-type ((Engine-Inlet-Cooling parameter)))
(causes (((increase AltitudeA)(decrease Engine-Inlet-Temp))
((decrease AltitudeA)(increase Engine-Inlet-Temp))
((increase AirspeedA)(increase Engine-Inlet-Temp))
((decrease AirspeedA)(decrease Engine-Inlet-Temp))
((increase MachA)(increase Engine-Inlet-Temp))
((decrease MachA)(decrease Engine-Inlet-Temp))
((increase Left-Hyd-Cooling-Pressure)(decrease Engi
ne-Inlet-Temp))
((decrease Left-Hyd-Cooling-Pressure)(increase Engi
ne-Inlet-Temp))
((increase Left-Elec-Cooling-Pressure)(decrease Eng
ine-Inlet-Temp))
((decrease Left-Elec-Cooling-Pressure)(increase Eng
ine-Inlet-Temp))
((increase Right-Hyd-Cooling-Pressure)(decrease Eng
ine-Inlet-Temp))
((decrease Right-Hyd-Cooling-Pressure)(increase Eng
ine-Inlet-Temp))
((increase Right-Elec-Cooling-Pressure)(decrease En
gine-Inlet-Temp))
((decrease Right-Elec-Cooling-Pressure)(increase En
gine-Inlet-Temp))))

(EngineA-Internal-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (EngineA-Internal-Cooling))
(association-type ((EngineA-Internal-Cooling parameter)))
(causes (((increase Fuel-FlowA)(increase EngineA-Interna
l-Temp))
((decrease Fuel-FlowA)(decrease EngineA-Interna
l-Temp))
((increase Left-Hyd-Cooling-Pressure)(decrease
EngineA-Internal-Temp))
((decrease Left-Hyd-Cooling-Pressure)(increase
EngineA-Internal-Temp))
((increase Left-Elec-Cooling-Pressure)(decrease
EngineA-Internal-Temp))
((decrease Left-Elec-Cooling-Pressure)(increase
EngineA-Internal-Temp))
((increase Right-Hyd-Cooling-Pressure)(decrease
EngineA-Internal-Temp))
((decrease Right-Hyd-Cooling-Pressure)(increase
EngineA-Internal-Temp))
((increase Right-Elec-Cooling-Pressure)(decreas
e EngineA-Internal-Temp))
((decrease Right-Elec-Cooling-Pressure)(increas
e EngineA-Internal-Temp))))

(EngineB-Internal-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (EngineB-Internal-Cooling))
(association-type ((EngineB-Internal-Cooling parameter)))
(causes (((increase Fuel-FlowB)(increase EngineB-Interna
l-Temp))
((decrease Fuel-FlowB)(decrease EngineB-Interna
l-Temp))
((increase Left-Hyd-Cooling-Pressure)(decrease
EngineB-Internal-Temp))
((decrease Left-Hyd-Cooling-Pressure)(increase

```

```

EngineB-Internal-Temp))
EngineB-Internal-Temp))
EngineB-Internal-Temp))
EngineB-Internal-Temp))
EngineB-Internal-Temp))
((increase Left-Elec-Cooling-Pressure)(decrease
EngineB-Internal-Temp))
((decrease Left-Elec-Cooling-Pressure)(increase
EngineB-Internal-Temp))
((increase Right-Hyd-Cooling-Pressure)(decrease
EngineB-Internal-Temp))
((decrease Right-Hyd-Cooling-Pressure)(increase
EngineB-Internal-Temp))
((increase Right-Elec-Cooling-Pressure)(decreas
EngineB-Internal-Temp))
((decrease Right-Elec-Cooling-Pressure)(increas
EngineB-Internal-Temp))))))

(Engine-Nozzle-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (Engine-Nozzle-Cooling))
(association-type ((Engine-Nozzle-Cooling parameter))))
(causes (((increase EgtA)(increase Engine-Nozzle-Temp))
((decrease EgtA)(decrease Engine-Nozzle-Temp))
((increase EgtB)(increase Engine-Nozzle-Temp))
((decrease EgtB)(decrease Engine-Nozzle-Temp))
((increase Left-Hyd-Cooling-Pressure)(decrease Eng
ine-Nozzle-Temp))
((decrease Left-Hyd-Cooling-Pressure)(increase Eng
ine-Nozzle-Temp))
((increase Left-Elec-Cooling-Pressure)(decrease En
gine-Nozzle-Temp))
((decrease Left-Elec-Cooling-Pressure)(increase En
gine-Nozzle-Temp))
((increase Right-Hyd-Cooling-Pressure)(decrease En
gine-Nozzle-Temp))
((decrease Right-Hyd-Cooling-Pressure)(increase En
gine-Nozzle-Temp))
((increase Right-Elec-Cooling-Pressure)(decrease E
ngine-Nozzle-Temp))
((decrease Right-Elec-Cooling-Pressure)(increase E
ngine-Nozzle-Temp))))))

(Vert-Tail-Temp temp-sensor (part-of (Thermal-Protection-SysA))
(associated-component (Vert-Tail-Cooling))
(association-type ((Vert-Tail-Cooling parameter))))
(causes (((increase AltitudeA)(decrease Vert-Tail-Temp))
((decrease AltitudeA)(increase Vert-Tail-Temp))
((increase AirspeedA)(increase Vert-Tail-Temp))
((decrease AirspeedA)(decrease Vert-Tail-Temp))
((increase MachA)(increase Vert-Tail-Temp))
((decrease MachA)(decrease Vert-Tail-Temp))
((increase Left-Hyd-Cooling-Pressure)(decrease Vert-Ta
il-Temp))
((decrease Left-Hyd-Cooling-Pressure)(increase Vert-Ta
il-Temp))
((increase Left-Elec-Cooling-Pressure)(decrease Vert-T
ail-Temp))
((decrease Left-Elec-Cooling-Pressure)(increase Vert-T
ail-Temp))
((increase Right-Hyd-Cooling-Pressure)(decrease Vert-T
ail-Temp))
((decrease Right-Hyd-Cooling-Pressure)(increase Vert-T
ail-Temp))
((increase Right-Elec-Cooling-Pressure)(decrease Vert-
Tail-Temp))
((decrease Right-Elec-Cooling-Pressure)(increase Vert-
Tail-Temp))))))

(Fwd-Hyd-Cooling-Pressure cooling-pressure (part-of (Thermal-Protection-SysA))
(associated-component (Fwd-Hyd-Cooling-Pump))
(association-type ((Fwd-Hyd-Cooling-Pump paramet
er))))

(Fwd-Elec-Cooling-Pressure cooling-pressure (part-of (Thermal-Protection-SysA))
(associated-component (Fwd-Elec-Cooling-Pump))
(association-type ((Fwd-Elec-Cooling-Pump param
eter))))

```

```

(Left-Elec-Cooling-Pressure cooling-pressure (part-of (Thermal-Protection-SysA))
  (associated-component (Left-Elec-Cooling-Pump)
    (association-type ((Left-Elec-Cooling-Pump par
      ameter))))))

(Right-Elec-Cooling-Pressure cooling-pressure (part-of (Thermal-Protection-SysA))
  (associated-component (Right-Elec-Cooling-Pum
    p))
    (association-type ((Right-Elec-Cooling-Pump p
      arameter))))

(Left-Hyd-Cooling-Pressure cooling-pressure (part-of (Thermal-Protection-SysA))
  (associated-component (Left-Hyd-Cooling-Pump))
    (association-type ((Left-Hyd-Cooling-Pump param
      eter))))

(Right-Hyd-Cooling-Pressure cooling-pressure (part-of (Thermal-Protection-SysA))
  (associated-component (Right-Hyd-Cooling-Pump)
    (association-type ((Right-Hyd-Cooling-Pump par
      ameter))))

```

Appendix C. *Users' Manual for the NASP System Status Monitor*

C.1 *File Structure*

The NASP System Status Monitor (SSM) is divided into separate program and data files distributed among seven subdirectories in the Symbolics 3600 file system. Each of the subdirectories has the prefix "host:>icat>alpha-demo," where "host" is the host name of the particular Symbolics computer on which the SSM is running. The seven subdirectories are:

1. packages
2. blackboard
3. interface
4. monitor
5. datafiles
6. stage1
7. stage2

Each of the seven subdirectories contains several files with pathnames of the form "host:>icat>alpha-demo>blackboard>x.y.z," where "x" is the filename, "y" is the file extension, and "z" is the file's version number.

There is one file used by the SSM which is not contained in one of the seven subdirectories. This file is "host:>icat>alpha-demo>alpha-demo-loader.lisp," and is used on start-up to load all the other program files from the seven subdirectories into the computer's memory.

C.2 *Installation*

If the NASP SSM is to be installed on a different Symbolics computer, all of the SSM program and data files must reside in the same subdirectories as listed in the preceding section. This is

required because the various files are specified by their full pathnames when they are loaded into memory.

C.3 Operation

The following steps should be taken to operate the NASP System Status Monitor.

C.3.1 Loading After booting the Symbolics computer, the following command should be typed at any Lisp Listener prompt;

```
(load ">icat>alpha-demo>alpha-demo-loader")
```

After typing the closing parenthesis, the command will be executed, and the SSM files will be loaded into the computer's memory. After all of the SSM files have been loaded, the Lisp Listener prompt will reappear.

C.3.2 Start-up After the SSM files have been loaded, the following command should be typed at the Lisp Listener prompt;

```
(blk:interact t)
```

This command will start the SSM in the interactive mode. The start-up process will take up to five minutes. When the start-up is complete, the NASP SSM System Display (Figure 9) will appear, with a mouse-sensitive menu in the lower righthand corner of the display.

C.3.3 Using the SSM The user interacts with the SSM by entering fault symptoms, starting the diagnosis and remediation functions, and then reviewing the results.

C.3.3.1 Entering Fault Symptoms To enter fault symptoms into the SSM, the user should use the mouse to choose the "Set Symptoms" selection in the "Interactive Monitor Menu." The next menu is the "Set Symptoms Menu," where the user should select "Add a Symptom." The next menu is the "Select Systems Menu" from which the user should select the aircraft system or

subsystem where the fault symptom is to appear. After selecting an aircraft system, that system's sensor menu will appear, and the user should select one of the sensors. Finally, the "Sensor Value Menu" will appear, and the user should select the desired value to apply to the selected sensor. After selecting a sensor value, the user will be returned to the "Set Symptoms Menu." From here, the user can either repeat the process listed above to enter more fault symptoms, or select "Quit" to return to the "Interactive Monitor Menu."

C.3.3.2 Starting the Diagnosis and Remediation Functions After symptoms have been set and the user has returned to the "Interactive Monitor Menu," the diagnosis and remediation functions are started by simply choosing the "Exit Monitor" selection. When these functions have completed, the current symptoms will be displayed in the Monitor Pane, and the results of the diagnosis function will be displayed in the Diagnosis Pane. If the diagnosis results cannot all fit in the Diagnosis Pane at one time, the word "*more*" will appear at the bottom of that pane. The user should press any key on the computer keyboard to display the next page of diagnosis information. This process is repeated until the last page of diagnosis information has been displayed and the "AirplaneA Menu" appears.

From the "AirplaneA Menu," the user can enter more or additional symptoms by selecting "Update," review the diagnosis or remediation results by selecting "Review Diagnosis," review one of the other aircraft system displays by selecting its name, or stop the SSM by selecting "Stop." If the user selects "Update" at this point and then adds more symptoms, the new symptoms will be added to the old symptoms. The diagnosis function will consider all the symptoms together, although it will only display the most recently added symptoms in the Monitor Pane. If the user selects "Initialize" before adding new fault symptoms, the previous symptoms will be deleted and only the most recently added symptoms will be considered by the diagnosis function.

Bibliography

1. B. Chandrasekaran et al. An approach to medical diagnosis based on conceptual structures. In *Proceedings of the Sixth IJCAI*, pages 134-142, 20-23 August 1979.
2. David Chihping Chen. *Hierarchical Plan Formation and Failure Recovery in the Flight Domain*. PhD thesis, Graduate College of the University of Illinois at Urbana-Champaign, Urbana, IL, 1985.
3. Randall Davis. Diagnosis via causal reasoning: paths of interaction and the locality principle. In *Proceedings of the National Conference on Artificial Intelligence*, pages 88-94, 22-26 August 1983.
4. Eugene L. Duke and Victoria A. Regenie. *Description of an Experimental Expert System Flight Status Monitor*. NASA Technical Memorandum 86791, Ames Research Center, Dryden Flight Research Facility, Edwards, California, May 1985.
5. L. D. Erman et al. The HEARSAY-II speech-understanding system: integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12:213-253, 1980.
6. Lee D. Erman et al. Engineering intelligent systems: progress report on ABE. In *Proceedings: Expert Systems Workshop*, pages 89-100, Science Applications International Corporation, McLean VA, April 1986.
7. Barbara Hayes-Roth. *BB1: An Architecture for Blackboard Systems that Control, Explain, and Learn about their own Behavior*. Heuristic Programming Project Report No. HPP-84-16, Stanford University CA, December 1984.
8. Barbara Hayes-Roth. *The Blackboard Architecture: A General Framework for Problem Solving?* Heuristic Programming Project Report No. HPP-83-30, Stanford University, CA, May 1983.
9. Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26:251-321, 1985.
10. Barbara Hayes-Roth. *A Blackboard Model of Control*. Heuristic Programming Project Report No. HPP-83-38, Stanford University, CA, August 1984.
11. Jay Lark. Module-oriented programming in ABE: modules and abstract datatypes. In *Proceedings: Knowledge-based Systems Workshop*, pages 7-19, Science Applications International Corporation, McLean VA, April 1987.
12. William Morris, editor. *The American Heritage Dictionary of the English Language*. Houghton Mifflin Company, Boston, 1979.
13. H. Penny Nii. Blackboard systems: blackboard application systems, blackboard systems from a knowledge engineering perspective. *The AI Magazine*, 82-106, August 1986.
14. H. Penny Nii. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. *The AI Magazine*, 38-53, Summer 1986.
15. Ramesh S. Patil et al. Causal understanding of patient illness in medical diagnosis. In *Proceedings of the IJCAI*, pages 893-899, 24-28 August 1981.
16. Harry E. Pople, Jr. The foundation of composite hypotheses in diagnostic problem solving: an exercise in synthetic reasoning. In *Proceedings of the IJCAI*, pages 1030-1037, 22-25 August 1977.
17. Harry E. Pople, Jr. et al. DIALOG: a model of diagnostic logic for internal medicine. In *Proceedings of the IJCAI*, pages 848-855, 3-8 September 1975.

18. Elaine Rich. *Artificial Intelligence*. McGraw-Hill Book Company, New York, 1983.
19. Paul C. Schutte and Kathy H. Abbott. *An Artificial Intelligence Approach to Onboard Fault Monitoring and Diagnosis for Aircraft Applications*. Technical Report. NASA Langley Research Center, Hampton, VA, 1986.
20. E. H. Shortliffe et al. An artificial intelligence program to advise physicians regarding antimicrobial therapy. *Computers and Biomedical Research*, 6:544-560, 1973.
21. Christopher A. Snyder. *A Parametric Study of a Gas-Generator Airturbo-Ramjet (ATR)*. NASA Technical Memorandum 88808, Lewis Research Center, Cleveland, Ohio, June 1986.
22. Kenneth E. Staten, Brigadier General, Program Manager for the National Aerospace Plane Program. Potential of the National Aerospace Plane. Speech to the 39th Annual Airport Operators International Convention, Denver, Colorado, 24 September 1986.
23. Donald A. Waterman. *A Guide to Expert Systems*. Addison-Wesley Publishing Company, Reading, MA, 1986.

Vita

Captain James M. Baumann was born on 23 January 1956 in Port Washington, Wisconsin. After graduating from Ozaukee High School, Fredonia, Wisconsin in 1974, Captain Baumann entered the United States Air Force Academy. He graduated with a Bachelor of Science degree in Electrical Engineering and received his USAF commission in May 1978. Captain Baumann's first Air Force assignment was as a radar systems engineer at Hanscom AFB, Massachusetts. He graduated from the USAF Test Pilots' School as a Flight Test Engineer in June 1983, and served as a Flight Test Engineer at Point Mugu Naval Air Station, California until entering the Air Force Institute of Technology, School of Engineering, in May 1986.

Permanent address: 2595 Highway I
Saukville, WI 53080

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GCE/ENG/87D-1			5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION School of Engineering	6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, OH 45433		7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION National Aerospace Plane JPO	8b. OFFICE SYMBOL (If applicable) AFSC/NAEF	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO 63269E	PROJECT NO	TASK NO
		WORK UNIT ACCESSION NO		
11. TITLE (Include Security Classification) A SYSTEM STATUS MONITOR FOR THE NATIONAL AEROSPACE PLANE				
12. PERSONAL AUTHOR(S) James M. Baumann, Captain, USAF				
13a. TYPE OF REPORT MS Thesis	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 1987 December	15. PAGE COUNT 128	
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP		
01	03	12	Artificial Intelligence Aerospaceplanes	
			Computer Aided Diagnosis Monitoring	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Thesis Chairman: Charles R. Bisbee, LtCol, USAF				
Approved for public release: IAW AFR 190.11. BY: [Signature] DATE: 14 Mar 88 Do not release to the public without Development AFIT/ENG/87D-1 (AFSC) Wright-Patterson AFB, OH 45433				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Charles R. Bisbee, LtCol, USAF			22b. TELEPHONE (Include Area Code) 513-255-2576	22c. OFFICE SYMBOL AFIT/ENG

The purposes of this study were to develop a model for an in-flight diagnostic system that could be applied to the National Aerospace Plane, and to implement a computer program to demonstrate the feasibility of that model as a basis for a system status monitor.

The diagnostic system model which was developed features a double hierarchy structure, one for the aircraft functions to be diagnosed, and another for the diagnostic functions to be performed. The hierarchical nature of both the system knowledge and the functions that use the knowledge allow decomposition of the diagnostic task into relatively independent and manageable parts.

The demonstration program which was developed includes a subset of the diagnostic system model. This program was implemented in Zetalisp on a Symbolics 3600 computer. It will simulate monitoring the dynamic performance parameters of an aircraft's subsystems, report any readings that fall outside of predetermined limits, reason about components responsible for the fault, display to the aircrew the other aircraft functions which may be affected by the component failure, and recommend actions that may remedy the fault situation.

The demonstration program clearly shows the validity of the diagnostic system model and highlights the importance of the causal and functional relationship techniques used to represent knowledge of the aircraft and its environment. The program demonstrates how the diagnostic system can supply relevant system status information to the aircrew. The report concludes with several recommendations for enhancements to the demonstration program.

END
DATE
FILMED

4-88
DTIC