A System Status Monitor for the National Aero-Space Plane

James M. Baumann and Charles R. Bisbee, III

Air Force Institute of Technology
Department of Electrical and Computer Engineering

Abstract

A study was undertaken to develop a model for an in-flight diagnostic system that could be applied to the National Aero-Space Plane, and to implement a computer program to demonstrate the feasibility of that model as a basis for a system status monitor.

The 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.

1 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 Aero-Space Plane (NASP). The NASP, flight vehicle, X-30, 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 X-30, will demonstrate this capability with only two crewmembers [13]. 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.

The problem investigated in this paper is to develop and demonstrate a strategy for an in-flight system status monitor for the National Aero-Space 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, report the possible consequences of the faults, and suggest remedial actions 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.

Since the National Aero-Space Plane is still in the planning stages, all references in this paper to the NASP experimental aircraft's capabilities and configurations are based on conjecture. However, these assumptions do not diminish the usefulness of the diagnostic model. In fact, the system status monitoring concept could be applied to any complex mechanical system where human operators have difficulty reacting in real-time to system anomalies.

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 the "Theoretical Development" section.

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 [5, 3]. Blackboards and their application to the NASP system status monitoring task will be discussed at length in the next section.

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 these were rejected in favor of a special purpose aircraft diagnosis system 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 [11, 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.

The System Status Monitor was tested with several sets of theoretical fault symptoms. The system performed well in most cases, but a number of areas needing improvement were discovered. These improvements are the subject of continuing research.

2 Problem Analysis

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

2.1 NASP Domain

The National Aero-Space 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 [13].

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 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.

2.2 Status Monitor Functions

As the name implies, a system status monitor should keep the flight crew appraised of the status of the aircraft systems. The first task (monitoring) is 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 is 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. The next section 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 must operate in the context of the aircraft as a collection of closely coupled, highly interactive systems.

2.3 Previous Work

The diagnostic process and the blackboard problem-solving model form the basis for NASP system status monitoring. Developmental work in these areas will be examined here.

Diagnosis is usually defined in medical terms as "the act or process of identifying or determining the nature of a disease through examination [6, 363]." In recent years, the meaning and application of diagnosis have been expanded to included 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" [14, 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."

Automated diagnosis was first applied in the medical field. One of the first and best-known medical diagnostic expert systems is MYCIN, 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 was developed at Stanford University. Work on this project by Shortliffe, Axline, Buchanan, Merigan, and Cohen was reported in the literature as early as 1973 [12].

Medical diagnosis systems that followed MYCIN, such as DIALOG [9], INTERNIST-II [8], ABEL [7], and MDX [1] moved away from simple rule-based approaches. These systems increasingly used "deeper" representations of system knowledge. This knowledge is "deep" in the sense that it represents how the system normally operates, rather than just a collection of specific instances of how the system fails.

Deep representation of system knowledge has carried over into automated hardware diagnostic systems. Notable in this area is the work of 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. It is this type of causal knowledge that has been incorporated in the NASP system status monitor model.

The other cornerstone of the NASP system status monitor model is the blackboard. 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 [4]. Since then, blackboards have been used in a wide variety of applications, and each time in a slightly different form [5, 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 after its necessary inputs have appeared in the blackboard.

The blackboard model is used in the NASP system status monitor to store the results of each of the individual diagnostic functions, and to serve as a communications path between those functions.

3 Theoretical Development

Development of the theory underlying the National Aero-Space Plane system status monitor will be covered in three parts in this section. 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.

3.1 Functional Hierarchy

The functional hierarchy, shown in Figure 1, was derived from the goal hierarchy developed by Schutte and Abbott [11], 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. 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 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).

3.2 Diagnostic Hierarchy

Although the diagnostic hierarchy is named for diagnosis, the actual diagnosis function is only one of five levels in the hierarchy. Figure 2 shows the diagnostic 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 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 numerical simulation of the normal 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.

To provide a meaningful input to the levels in the diagnostic hierarchy which use symbolic processing, the monitor must 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.

Ideally, the diagnosis function should identify 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.) 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.

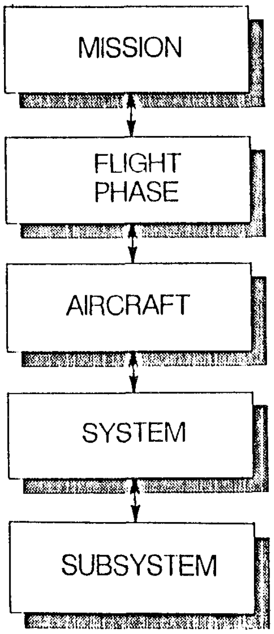

Figure 1: Functional Hierarchy.

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.

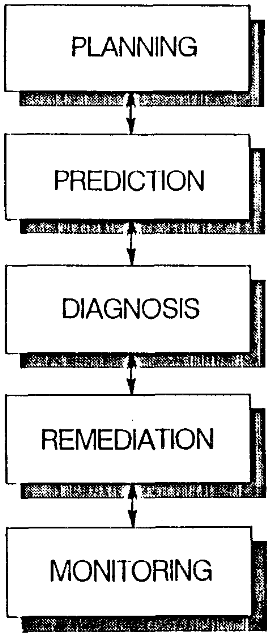

Figure 2: Diagnostic Hierarchy.

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.

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

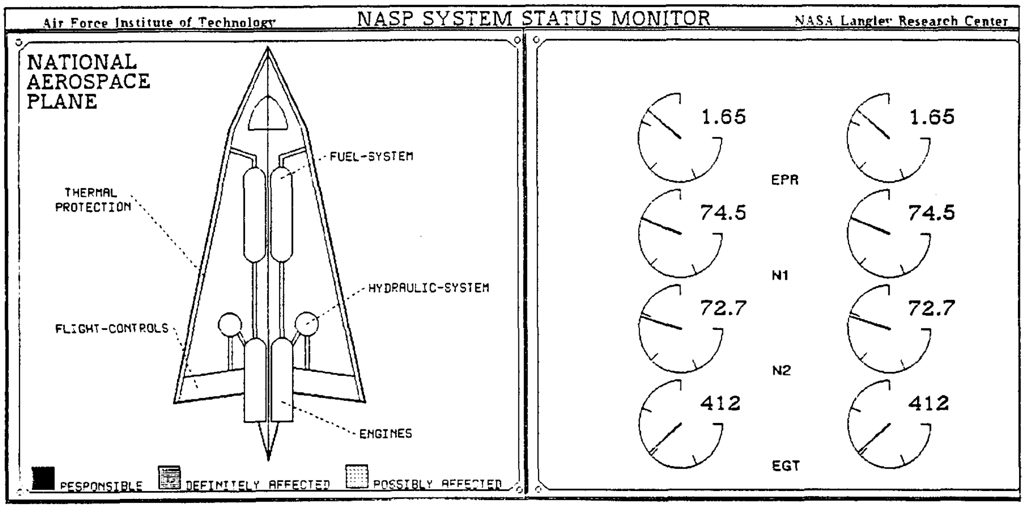

Figure 3: NASP SSM System Display.

in the NASP system status monitor. It was decided the remediation function should seek to compensate for the current set of fault symptoms in order to ensure mission accomplishment. Implementation of other remediation approaches is the object of further research.

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. 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. 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.

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. The proposed remedy may have 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.

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.

The planning function will determine the long-range consequences of the proposed remedial action. The planning function must determine if the consequences of the remedial action will allow completion of the mission. If the proposed remedy is consistent with the mission objectives, it 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. The planning function must pick the best remedy while working within any constraints, such as safety, cost, security, etc.

3.3 Semantic Network Knowledge Representation

The knowledge about the components and functions of the National Aero-Space Plane are organized in a semantic network representation. Semantic networks were originally developed as a way of representing the meaning of English words [10, 215]. The objects to be represented are the nodes of the network, and the relationships between the objects are the links connecting the nodes. Each link has a direction to signify the direction of the relationship. Two-way relationships must be expressed explicitly.

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 links 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. In addition to the "PARTS" links, the most important links in the NASP semantic network knowledge base are the "FUNCTIONAL-DEPENDENTS" links, which show that the proper functioning of one component depends on the proper operation of other components.

3.4 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 in the knowledge base attached to the airspeed sensor 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. 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 this study, alternative 2 was used to select the most appropriate remedial action.

4 Prototype Development

A prototype System Status Monitor (SSM) was implemented using the theory developed in the previous section.

The four major components of the SSM are structured around the functional and diagnostic hierarchies discussed earlier. These components are the;

1. Semantic network knowledge base,
2. Monitoring function,
3. Diagnosis function, and
4. Remediation function.

The semantic network knowledge base is organized according to the functional hierarchy, as shown in Figure 1. The monitoring, diagnosis, and remediation functions are the lower three levels of the diagnostic hierarchy, shown in Figure 2.

4.1 Knowledge Base

A general framework of the NASP aircraft and its functions are represented in the knowledge base. Below the top, or "Mission," level of the functional hierarchy, the knowledge base contains information on the five primary flight phases (takeoff, climb, cruise, descent, and landing). At the system level, five of the major aircraft systems are represented. These systems are propulsion, hydraulics, fuel, flight controls, and thermal protection. Each system may have one or more subsystems, and the subsystems have various constituent components. While the hierarchical nature of the knowledge base provides orderliness and structure to the knowledge, it is the links between the objects in the knowledge base that are used by the other SSM functions. The links which are most extensively used by the diagnosis function are the "FUNCTIONAL-DEPENDENTS" links, while the remediation function uses the "CAUSES" links.

4.2 Monitoring Function

When the SSM is started, the user is presented with the computer display shown in Figure 3 and is in the monitoring function. The user interactively selects fault symptoms from a list of 54 possible symptoms. The symptoms are associated with abnormal readings from sensors in each of the five aircraft systems, and from sensors that measure overall aircraft performance, such as the altimeter and vertical velocity indicator. When the desired set of fault symptoms have been chosen, the user exits the monitoring function. The current fault symptoms are printed in the lower left portion of the computer display and the system automatically enters the diagnosis function.

4.3 Diagnosis Function

The SSM performs a two-stage diagnosis function and displays its results in both text and graphics form. These results show both the component suspected of being the cause of the current fault symptoms, and the other aircraft components or functions which may be affected by the current fault situation.

Stage 1 of the diagnosis function compares the current symptoms to a set of stored fault-symptom association rules. If the current fault symptoms match the "IF" part of a rule, the "THEN" part of the rule is invoked, naming the cause of the fault. This 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.

Stage 2 performs its version of the diagnosis function by producing a series of diagnosis hypotheses. These hypotheses start with one of the lowest-level (or primitive) components in the functional hierarchy. Each hypothesis is based on the assumption that the primitive component upon which that hypothesis is built is somehow responsible for all the current fault symptoms. With this assumption, a hypothesis is built by exhaustively searching through the knowledge base, following the "FUNCTIONAL-DEPENDENTS" links from the primitive component to the other components which are so linked. These "fault propagation paths" are then compared to the current fault symptoms. If all the current symptoms lie on the propagation path, the hypothesis is valid. If the hypothesis' propagation path does not explain each of the current symptoms, that hypothesis is discarded. If none of the hypotheses is valid, Stage 2 produces a default hypothesis.

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. Each component in a valid hypothesis' propagation path is listed with its assigned fault severity. A "RESPONSIBLE-COMPONENT" is the primitive 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 results of a fault hypothesis are also displayed graphically in the upper left portion of the display, called the system window. There are 16 different displays that can be shown in the system window. These displays represent, in pictorial form, all or part of one of the five levels 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 user (or the flight crew, if this system were actually installed in the aircraft) those components affected by a fault situation.

4.4 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 earlier, the remediation function would seeks 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 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." This process continues, producing a string of actions that should ultimately counteract the affects of the highest-level fault symptom. The results of the remediation function are also displayed in the lower right portion of the system display.

5 Conclusions

The System Status Monitor has been tested with a variety of fault symptom sets. The results of the test runs show that the 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. Based on the results of the diagnosis function, the user is presented with a display of the system components affected by the fault, and is given suggested actions to remedy the fault situation.

Future work on this project is being considered in the following areas:

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 raw sensor readings to produce an event-driven simulation of a NASP mission.

2. 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.

References

[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] L. D. Erman et al. The HEARSAY-II speech-understanding system: integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12:213–253, 1980.

[5] 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.

[6] William Morris, editor. *The American Heritage Dictionary of the English Language.* Houghton Mifflin Company, Boston, 1979.

[7] 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.

[8] 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.

[9] 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.

[10] Elaine Rich. *Artificial Intelligence.* McGraw-Hill Book Company, New York, 1983.

[11] 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.

[12] E. H. Shortliffe et al. An artificial intelligence program to advise physicians regarding antimicrobial therapy. *Computers and Biomedical Research*, 6:544–560, 1973.

[13] 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.

[14] Donald A. Waterman. *A Guide to Expert Systems.* Addison-Wesley Publishing Company, Reading, MA, 1986.