Robust Operative Diagnosis as Problem Solving in a Hypothesis Space

Kathy H. Abbott
NASA Langley Research Center
Hampton, Virginia 23665-5225

Abstract

The lack of robustness in current diagnostic systems is an important research issue because it has two major consequences: inability to diagnose novel faults and inability to diagnose more than one type of fault. This paper describes an approach that formulates diagnosis of physical systems in operation (operative diagnosis) as problem solving in a hypothesis space. Such a formulation increases robustness by: (1) incremental hypotheses construction via dynamic inputs, since the fault propagation results in changes in symptoms over time; (2) reasoning at a higher level of abstraction to construct hypotheses, albeit less specific ones, when specific knowledge is not available; and (3) partitioning the space by grouping fault hypotheses according to the type of physical system representation and problem solving techniques used in their construction. The approach was implemented for aircraft subsystems and evaluated on eight actual aircraft accident cases involving engine faults, with very promising results.

1 Introduction

The lack of robustness in current diagnostic systems is an important research issue because it has two major consequences: inability to diagnose novel faults and inability to diagnose more than one type of fault. For example, most current approaches to diagnosis depend on compiled, specific knowledge about the associations between symptoms and faults. However, when novel faults occur for which there is no specific associational knowledge, approaches that depend on such knowledge are inadequate. When the diagnosis is done for physical systems in operation (*operative diagnosis*), it is even more important to diagnose novel faults because the cost of inappropriate responses may be high.

The purpose of operative diagnosis is to facilitate continued, safe operation, rather than identifying the part to repair. Moreover, identifying the *effects* of the fault on the status of the physical system is equally as important as identifying the *cause* of the fault. In operative diagnosis, determining system status is often a dynamic process, as the effect of the fault propagates while the system continues to operate. Therefore, the operative nature of the diagnosis affects the reasoning in two ways: the need to reason about dynamic inputs and to generate system status. Another important consideration is that testing for additional information is limited because of the need for safe, continued operation. Limited testing means that information available to discriminate hypotheses is less than sometimes desired. Moreover, sensed parameters are not available for every component in the system, and these sensor readings are sampled at (usually fixed) intervals. The set of symptoms may change because of fault propagation, and some changes may be undetected between samples.

Much research has been done in diagnosis. Several of these approaches diagnose known faults where the effect of the fault propagates. For example, [Fagan *et al.*, 1984; Patil, 1987; Weiss *et al.*, 1978; Pan, 1983] address diagnosis of known faults. Although these and other research efforts address the problem in much depth, they do not address novel faults.

The fragility of these systems motivated several current approaches that use deep models in the diagnosis process [Fink and Lusth, 1987; Davis, 1985; Hamilton *et al.*, 1986]. These model-based approaches generate hypotheses that identify the cause of the problem, (e.g., the faulty component), but not the system status. While this may be sufficient in cases where all the diagnostician needs to do is identify the part to replace, it assumes that no other parts need to be replaced or repaired as a result of the fault. Additionally, because they only use functional models, they cannot diagnose failures where one component damages another physically-adjacent component. Their capability to use multiple physical system representations is limited or nonexistent. Diagnosing some faults requires multiple representations [Davis, 1985], although even Davis' approach cannot generate system status or combine representations.

This paper describes an approach that views diagnosis as problem solving in a hypothesis space. This view enables an improvement in robustness through *incremental hypothesis updates*, and the *abstraction* and *partitioning* of the hypotheses in the space. *Incremental hypothesis updates* enable diagnosis of dynamic fault behavior caused by fault propagation. Within the hypothesis space, the approach uses specific associational knowledge when available. However, when novel faults occur, the diagnostic problem solver uses *abstraction of the individual hypotheses* to provide a diagnosis, albeit a less specific diagnosis. The hypothesis space is *partitioned into fault classes*, grouping faults into different classes if their behavior requires different problem solving techniques or representations to diagnose them. Other diagnostic approaches can be viewed as diagnosing a subset of the classes included here.

This approach was implemented in a computer program called Draphys (*D*iagnostic *R*easoning *A*bout *Phys*ical *S*ystems) and demonstrated in the domain of aircraft sub-

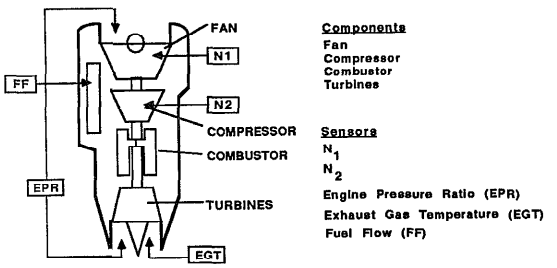

Figure 1: Aircraft Turbofan Engine.

systems; specifically, an aircraft turbofan engine and hydraulic subsystem. The approach was evaluated using actual aircraft accident cases involving engine faults, with very promising results.

2 Hypotheses in Operative Diagnosis

In operative diagnosis, the diagnosis is done to assist in continued operation of the system under consideration. Each element of the diagnosis problem space is a hypothesis that describes the cause of the fault and the current system status.

In Draphys, a hypothesis includes: the *fault type*, the *cause* or *source* of the problem, the *propagation path*, and the *system status*. The fault type is either single fault or multiple independent faults. The source of the fault is the physical component that is broken. The specific cause describes **how** that component is broken. The propagation path describes the order in which the fault affected the components. The system status describes the components affected by the fault and their operational status. The operational status of an affected component is either *definitely affected* by the failure when symptom information justifies it, or *possibly affected* when there is reason to believe that the component might be affected but symptom information cannot confirm or refute it.

3 Diagnosis as Problem Solving in a Hypothesis Space

3.1 Aircraft Subsystem Diagnosis

Inflight diagnosis of aircraft subsystems is an example of operative diagnosis. The aircraft subsystems diagnosed by Draphys are two turbofan engines, two fuel subsystems, and a hydraulic subsystem. A schematic of the engine used in later examples is shown in figure 1.

The input to the diagnosis system is a set of qualitative sensor values that identify which sensors are abnormal and how they are abnormal, e.g., fuel flow is high. A fault monitor generates these symptoms by comparing the sensor readings to expected values computed from a numerical simulation model of, for example, the engine. Schutte [Schutte and Abbott, 1986] describes the fault monitor.

3.2 Diagnosis of Known Faults

Draphys uses compiled associational knowledge to diagnose known, commonly occurring faults. For the aircraft domain, the fault-symptom associations were obtained by interviewing domain experts (pilots and engine designers) and by examining actual fault cases. They were implemented in a rule-based system that permits the temporal functions defined by Allen [Allen, 1984] as part of the rules.

These rules can adequately capture the sequence of symptoms as described by the experts, but have representational limitations as discussed in [Abbott *et al.*, 1987]. These include the awkwardness of expressing all the propagation behavior over time that could occur for any one fault, as well as temporal duration. The major question addressed below is what to do if the fault is one whose symptoms do not correspond to the associational knowledge. This question is of great interest, since novel faults appear to be very difficult for humans to diagnose.

3.3 Graceful Degradation Via Abstraction

Much of the related research views graceful degradation in the presence of novel faults as reasoning with deep models. In such a view, it is the *efficiency* of the reasoning process that degrades. The approach presented here does not view graceful degradation as an issue of degraded efficiency, but an issue of degraded specificity. If the diagnostic system cannot identify exactly what the fault is, it can still generate useful diagnostic information, even if that information is less specific than desired.

Before presenting the approach, it is useful to explore what information should be abstracted and why. If the goal of the diagnostician is to select a remedial action to take in response to the fault, information should be generated to support that selection. During the interviews of experts, they described default actions that they would take if they did not recognize the fault or if there were multiple hypotheses. This action was generally a conservative response to the fault. For example, if the pilot knew he had a compressor failure, but did not know how the fan was broken, he would shut down the engine. However, if he knew it was an eroded compressor blade, he might reduce the throttle on that engine. Thus he had an action associated with the general class of compressor failures that was (potentially) different from the action associated with the specific compressor fault.

Motivated by this and other examples, a structured way of forming general categories of faults with associated default actions was identified. In the aircraft domain, these categories are defined as the components in the physical system, as exemplified above. When novel faults occur, diagnostic reasoning takes place at a higher level of abstraction. Hypotheses are produced that identify what component is faulty, without identifying how the component is broken. The operational status of the component that is abstracted (e.g., abnormal rather than low pressure), so this abstraction is called *status abstraction*. Draphys uses two such levels, shown in figure 2.

Since the diagnostic reasoning at the higher abstraction level is designed to identify the component that is faulty, the symptoms can be abstracted as well. Although it is

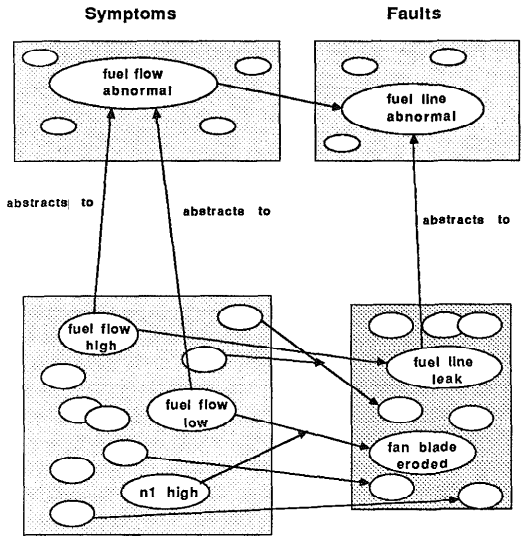

Figure 2: The Two Levels of Status Abstraction in Draphys.

necessary at the lower level to identify how the symptomatic sensor compares with its expected value (e.g., high or low), it is not necessary to make this distinction at the higher level. It is only necessary to identify that the value of the sensor is abnormal. This is also status abstraction, but it is the parameter value status that is abstracted. Figure 2 also illustrates the relationship between the specific fault hypotheses and the corresponding symptoms.

The reasoning at the higher level of abstraction is a generate-and-test process. When symptoms first appear, the generator localizes the fault in a component hierarchy, resulting in a set of candidate components that might be the source of the problem. It then constructs fault hypotheses by simulating fault propagation from each of the candidates. Each resulting hypothesis is then tested to determine if it is valid; that is, if it accounts for all the current symptoms. Often this generate-and-test process results in multiple valid hypotheses. If new symptoms arrive as time progresses, the generator incrementally updates the old hypotheses to determine whether they can account for the new symptoms. If they can, the generator retains them. Otherwise, it prunes them.

An example will clarify this process. Suppose the fault is a fan failure. In such a failure, the first sensor affected would be N_1. Since the fan would not compress air properly, the effect of that failure would propagate to the high-pressure compressor and thus to N_2. It would then propagate to the combustor since the under-compressed air would not ignite as efficiently. Therefore, the expanding gases resulting from combustion would not turn the turbines as rapidly as it normally would. EGT and EPR

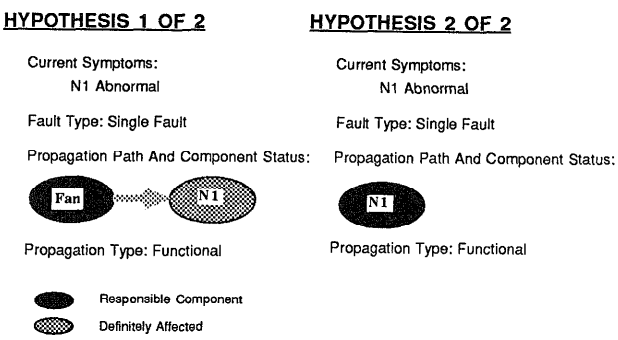

Figure 3: Hypotheses Resulting From a Symptom in N_1.

would be symptomatic to reflect this. The turbines would not be extracting energy, so the fan and compressor would not turn as fast since they derive their power from the turbines. Thus the faulty response is perpetuated.

For this fault, suppose that the first symptom that Draphys detects is in N_1. Since N_1 is an engine parameter, Draphys is able to localize the fault to the engine subsystem. Each component in the engine subsystem is then proposed as a candidate responsible component.

For each proposed responsible component, Draphys generates a fault hypothesis by qualitatively simulating the fault propagation behavior. For example, when Draphys proposes the fan as the responsible component, it uses a model of the engine and its functional interconnections to determine that the high-pressure compressor and the N_1 sensor functionally depend on the fan. Knowing these interconnections, Draphys then attempts to continue simulating the propagation of the failure to these functionally dependent components. In this example, it checks whether the fault's effect has reached the high-pressure compressor by examining the symptoms to see if N_2 is symptomatic. If it is, then Draphys assumes that the failure affects the high-pressure compressor, and continues the process from there. If N_2 is not symptomatic, as in this example, simulated propagation halts on this path. Draphys then explores all remaining functional propagation paths. After exhausting all paths, the hypothesis is tested for validity.

Draphys does the same process for each candidate component. In this example, two valid hypotheses are generated, shown in figure 3. The first is that the fan is the responsible component, and the second is that the N_1 sensor failed. A fault in either component could result in the current symptoms.

Extending this example illustrates the incremental updating of hypotheses. Assume that a short time after the N_1 symptom was first detected and diagnosed, a symptom in N_2 is detected. Draphys then tries to extend the propagation path of all the valid hypotheses to explain the new symptoms by continuing the qualitative simulation from the end of the propagation path in the old hypotheses. For instance, in one valid hypothesis propagation stopped at the fan, because the next component on this functional propagation path was the high-pressure compressor. Since earlier there was no symptom in N_2, Draphys assumed that the compressor was unaffected. Now that there is a

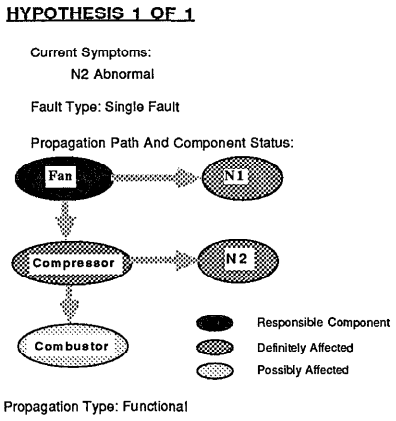

Figure 4: Hypothesis Remaining After a Symptom in N_2.

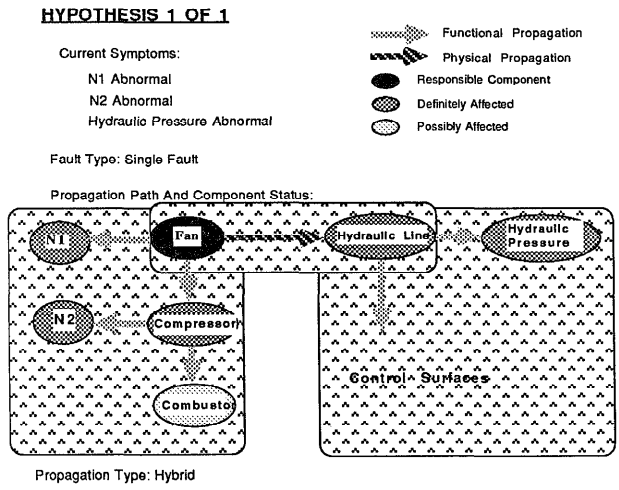

Figure 5: Composed Hypothesis Explaining a Fan Blade Separation.

symptom in N_2, Draphys updates the system status for this hypothesis and continues the simulated propagation.

The resulting hypothesis accounts for all symptoms. It is the only member of the set of old valid hypotheses that can do so, since a sensor failure in N_1 could not result in functional propagation that would account for the symptom in N_2. Figure 4 shows this remaining hypothesis.

3.3.1 Using Multiple Physical System Representations

The reasoning based on the functional model is sufficient for the faults that propagate along functional dependency links, but not all faults do. Suppose that the fan blade broke off and damaged a hydraulic line in the wing to which the engine was attached. The monitor detects symptoms in N_1 and in the hydraulic pressure sensor. Draphys cannot explain these symptoms by simulating functional propagation, because there is no functional relationship between these components.

A physical proximity relationship does exist. Therefore, by knowing that the fan is physically adjacent to the wing containing the hydraulic line, Draphys can identify propagation from the engine to the wing. This represents another class of faults, since it requires a different representation (physical rather than the functional structure). This type of fault is analogous to Davis' bridge fault [Davis, 1985].

The reasoning process used is the same as described with faults that propagate functionally, except that the models used in localization and simulation are based on physical structure rather that functional structure. The component hierarchy used for localization groups components according to physical location rather than functional relationships. The simulation model used is a specialized model of physical proximity. This specialized model includes directional information in representing these physical proximity relationships. For instance, it is possible for the fan blade to break off and damage the hydraulic line, but not vice versa.

Unfortunately, reasoning with a single representation is not sufficient. Once a fan blade separation has caused damage in both the engine and in the hydraulic system, the effect of the fault will propagate functionally in both subsystems. The initial propagation was physical, but subsequent propagation was functional. Therefore, explaining the current fault behavior requires models of both physical and functional structure. Draphys diagnoses hybrid fault propagation by composing the simple hypotheses that describe the single type of propagation, as illustrated in figure 5.

Faults involving physical damage illustrate that some known faults are more appropriately represented at the higher level of abstraction. The reasoning described for diagnosing physical damage could be compiled into specific rules, but doing so may not improve ability to take remedial action. Moreover, physical damage can occur so many different ways that a large number of specific rules would result, possibly inhibiting their timely retrieval.

3.3.2 Partitioning the Hypothesis Space

So far, four fault classes were described that require different problem solving techniques or different physical system representations. Figure 6 includes these four fault classes, and shows the partitioning of the hypothesis space. The present implementation of Draphys diagnoses all fault classes shown except for multiple faults. The fault classes are examined in order of likelihood and correspond to a depth-first, left-to-right traversal of the space as shown.

4 Evaluation

Draphys was evaluated by reconstructing actual civil transport aircraft accident cases and using their symptoms as input [a; b; c; d; e; f; g; h]. Each accident was an engine-related failure that resulted in the loss of life and property. Four of the eight accident cases were used to guide the design and construction of Draphys. The remaining four were set aside for evaluation purposes. All eight were reconstructed by an objective party and presented as input

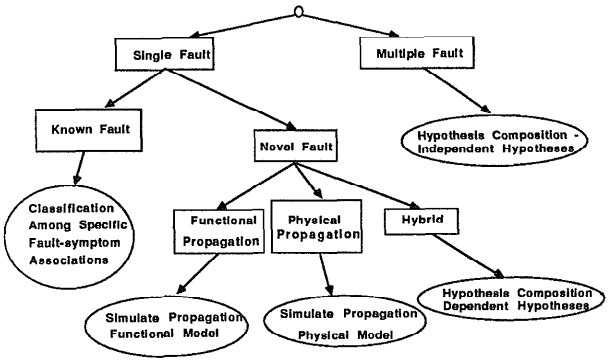

Figure 6: Hypothesis Space Partitioning.

to Draphys[1]. Each level of abstraction was invoked for each case to determine the diagnosis success of the associational rules at the specific level and the generate-and-test at the higher abstraction level. The physical system model used contained approximately 40 components and 100 interconnections. A brief summary of the resulting hypotheses (without system status) is shown in table 1.

A successful diagnosis was defined as one in which the correct hypothesis was among the set of valid hypotheses. This definition was used because Draphys may generate several valid hypotheses for a particular set of symptoms. It may be impossible to isolate to one hypotheses with the sensor information available, even for a human expert. Moreover, since Draphys does not yet include any representation of uncertainty, the valid hypotheses cannot be ordered by likelihood.

Using this criterion for success, seven of the eight accident cases were successfully diagnosed. Of the seven successes, two were diagnosed using the associational rules at the specific level of abstraction. All seven of the successes were diagnosed at the higher abstraction level. Of these seven cases, five involved physical damage. In each of the five cases, functional propagation resulted from the physical damage. No physical damage cases were diagnosed successfully by the associational rules at the specific abstraction level.

The accident case that was not successfully diagnosed was not a structural fault. It involved massive water ingestion into the engine during a heavy rainstorm, leading to engine failure. Modifying Draphys to diagnose this failure would require modeling the inputs to a device as a potential source of the fault, which may be a desirable enhancement.

Why did this approach work so well? The credit for success lies mainly with two aspects of the approach: the

[1]I am indebted to Paul Schutte for reconstructing the accident cases and doing the initial evaluation as described in [Schutte *et al.*, 1987].

Case Description	Stage 1 Hypothesis	Stage 2 Hypothesis
1. Turbine Blade Separation	*1. Turbine Blade Separation 2. Flameout	1. Fan 2. Compressor 3. Combustor * 4. Turbine
2. Fan Failure	1. Turbine Blade Separation	* 1. Fan
3. Fan Failure	1. Turbine Blade Separation	* 1. Fan
4. Foreign Object Ingestion	none	* 1. Fan 2. Compressor 3. Combustor 4. Turbine
5. Water Ingestion	1. Turbine Blade Separation 2. Flameout	1. Combustor 2. Turbine
6. Engine Separation	1. Fuel System Failure 2. Flameout	*1. Engine - Fan
7. Turbine Disk Separation	* 1. Turbine Blade Separation	1. Combustor * 2. Turbine
8. Bearing Failure	1. Turbine Blade Separation 2. Flameout	*1. Compressor *correct diagnosis

Table 1: Summary of Accident Case Diagnoses.

symptoms detected and the models used. Symptoms provided by the monitor identify abnormal sensor readings as soon as they occur (or the first sample thereafter). This detects symptoms sooner than current operational systems, which alert the operator when a sensor exceeds its total normal operating range.

The physical system models used must represent the behavior of the faulted system for the fault class being diagnosed. For example, the functional model represents a model of the normal system, but is at a high enough level of abstraction that it represents behavior under many fault conditions as well. In contrast, the model of physical structure only includes directional proximity information for possible physical damage, thus it does not model normal behavior. Including all nondirectional proximity relationships may be much less efficient and might not incorporate domain knowledge known from the device design about how internal physical damage might occur. In addition to appropriate representations, the ability to combine the physical and functional models was also important.

5 Conclusions

This paper presented an approach that views diagnosis as problem solving in a hypothesis space. With this view, robustness is improved through reasoning about fault propagation, permitting incremental hypothesis construction; status abstraction of the individual hypotheses, and partitioning of the hypothesis space to group fault hypotheses according to representation and problem solving technique.

Incremental hypothesis construction based on fault propagation behavior can be used to discriminate hypotheses, particularly when symptoms change over time. How-

ever, this requires that the detection process identify when sensor readings become abnormal, not just when they exceed the normal operating range.

Abstraction of hypotheses is useful when actions are associated with the general fault categories represented by the abstract hypotheses. The approach of using different abstraction levels for diagnosing novel faults is appropriate when specific hypotheses are most desirable, but abstract hypotheses are better than nothing. Moreover, some known faults are more appropriately represented at the higher level of abstraction, such as, physical damage. This is the case when more specific hypotheses do not improve ability to take remedial action or the increase in number of specific hypotheses would inhibit their timely retrieval.

Partitioning the fault space is appropriate when different problem solving techniques or representations are required to diagnose different classes of faults. In this approach, different fault classes and their corresponding diagnostic techniques and representations were identified. One of these classes involved diagnosis of faults which propagate within multiple representations, which no other current approach can do. Evaluation of this approach revealed that diagnostic capability depends on the available physical system models and the fault propagation behavior that they can represent.

Acknowledgements

The research described here is part of the author's dissertation research at Rutgers University. I thank my advisor, Professor Lou Steinberg, and Professors Chris Tong, Don Smith, and Chuck Schmidt for guidance and suggestions. Peter Friedland, Paul Schutte, and George Steinmetz also commented on a draft of this paper.

References

[Abbott *et al.*, 1987] K. Abbott, P. Schutte, M. Palmer, and W. Ricks. Faultfinder: a diagnostic expert system with graceful degradation for onboard aircraft applications. In *14th International Symposium on Aircraft Integrated Monitoring Systems*, Friedrichshafen, West Germany, September 1987.

[Allen, 1984] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23, 1984.

[Davis, 1985] Randall Davis. Diagnostic reasoning based on structure and function. In Daniel G. Bobrow, editor, *Qualitative Reasoning About Physical Systems*, The MIT Press, 1985.

[Fagan *et al.*, 1984] L. Fagan, J. Kunz, E. Feigenbaum, and J. Osborn. Extensions to a rule-based formalism for a monitoring task. In B. Buchanan and E. Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, 1984.

[Fink and Lusth, 1987] P. K. Fink and J. C. Lusth. Expert systems and diagnostic expertise in the mechanical and electrical domains. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-17(3), 1987.

[Hamilton *et al.*, 1986] T. Hamilton, D. Simmons, and R. Carlson. HELIX: an engine monitoring system. In *41st Mechanical Failure Preventions Group Symposium*, Oct 1986.

[Pan, 1983] Y-C. Pan. *Qualitative Reasonings With Deep-Level Mechanism Models for Diagnosis of Dependent Failures.* PhD thesis, University of Illinois, 1983.

[Patil, 1987] R. Patil. A case study on evolution of system building expertise: medical diagnosis. In Grimson and Patil, editors, *AI in the 1980s and Beyond*, MIT Press, 1987.

[Schutte and Abbott, 1986] P. Schutte and K. Abbott. An artificial intelligence approach to onboard fault monitoring and diagnosis for aircraft applications. In *AIAA Guidance, Navigation, and Control Conference*, 1986.

[Schutte *et al.*, 1987] P. Schutte, K. Abbott, M. Palmer, and W. Ricks. An evaluation of a real time fault diagnosis expert system for aircraft applications. In *Proceedings of the 26th IEEE Conference on Decision and Control*, 1987.

[Weiss *et al.*, 1978] S. M. Weiss, C. Kulikowski, S. Amarel, and A. Safir. A model-based method for computer-aided medical decision making. *Artificial Intelligence*, 11:145–172, 1978.

[a] *Aircraft Accident Report: United Airlines, Inc., Boeing 737-222, N9005U, Philadelphia International Airport, Philadelphia, Pennsylvania, July, 19, 1970.* National Transportation Safety Board. NTSB-AAR-72-9.

[b] *Aircraft Accident Report: National Airlines, Inc., DC-10-10, N60NA, Near Albuquerque, New Mexico, November 3, 1973.* National Transportation Safety Board. NTSB-AAR-75-2.

[c] *Aircraft Accident Report: Overseas National Airways, Inc., Douglas DC-10-30, N1032F, John F. Kennedy International Airport, Jamaica, New York, November 12, 1975.* National Transportation Safety Board. NTSB-AAR-76-19.

[d] *Aircraft Accident Report: Southern Airways Inc., DC-9-31, N1335U, New Hope, Georgia, April 4, 1977.* National Transportation Safety Board. NTSB-AAR-78-3.

[e] *Aircraft Accident Report: American Airlines Inc., DC-10-10, N110AA, Chicago-O'Hare International Airport, Chicago, Illinois, May 25, 1979.* National Transportation Safety Board. NTSB-AAR-79-17.

[f] *Aircraft Incident Report: Northwest Airlines 79, McDonnell Douglas DC-10-40, N143US, Leesburg, Virginia, January 31, 1981.* National Transportation Safety Board. NTSB-AAR-81-10.

[g] *Aircraft Accident Report: Air Florida Airlines, Inc., McDonnell-Douglas, Inc. DC-10-30CF, N101TV, Miami, Florida, September 22, 1981.* National Transportation Safety Board. NTSB-AAR-82-3.

[h] *Aircraft Accident Report: Eastern Airlines Flight 935, Lockheed L-1011-384, N309EA, Near Colts Neck, New Jersey, September 22, 1981.* National Transportation Safety Board. NTSB-AAR-82-5.