

# Cases, Models, and Rules for Fault Management: An Evaluation

**Stamos T. Karamouzis<sup>1</sup>**  
NASA Langley Research Center  
Mail Stop 152  
Hampton, VA 23681-0001  
stamos@icat.larc.nasa.gov

**Stefan Feyock<sup>2</sup>**  
Department of Computer Science  
College of William & Mary  
Williamsburg, VA 23185  
feyock@cs.wm.edu

## Abstract

This paper describes an experimental evaluation of the diagnostic concepts implemented in a prototypical case-based reasoner (CBR), called Epaion. The reasoner operates in the domain of in-flight fault diagnosis and prognosis of aviation subsystems, particularly jet engines. The evaluation used actual aircraft accidents and incident cases, which were simulated to assess the effectiveness of Epaion in diagnosing failures. Results of this evaluation, together with a brief description of Epaion, are presented. Additionally, these results are compared with the results achieved by the rule-based and the model-based processes of another reasoner which operates in the same domain.

## Introduction

The current state of knowledge-based systems offers a variety of reasoning methodologies and tools. Rule-based and model-based reasoning are the most popular reasoning paradigms used in a variety of domains. On the other hand, recent advances in Case-based reasoning (CBR) research have moved CBR from the research bench into the applications arena at a rapid pace.

Reasoning about physical systems is a difficult process, and any attempt to automate this process must overcome a number of challenges. Among these are the tasks of generating explanations of normal behavior, fault diagnoses, explanations of the various manifestations of faults, prediction of future behavior, etc. The reasoning process becomes even more difficult when physical systems must remain in operation. During operation, a physical system is changing dynamically by modifying its set of components, the components' states and pattern of interconnections, and the system's behavior.

To address these concerns a research effort has been initiated at NASA/Langley Research Center aiming at the study, design, and development of AI-based systems for in-flight fault management (Abbott 1991, Karamouzis &

Feyock 1993a). The research deals with the domain of in-flight fault diagnosis and prognosis of aviation subsystems, particularly jet engines. Automation of in-flight fault diagnosis and prognosis can be used as an aid to the flight crew for early detection of a problem or failure. This provides the crew with more time to respond more effectively and reduce potential damage due to the failure.

The research effort at Langley produced two reasoners. The first reasoner, called DRAPHYS, performs a two-fold reasoning process, one that is based on a set of rules and the other that is based on models of the aircraft. The second reasoner, called Epaion, involves the use of case-based techniques in conjunction with models that describe the aircraft.

This paper is an extension of (Karamouzis and Feyock 1993b) that described an evaluation of the diagnostic concepts implemented in Epaion. The evaluation used reports of actual aircraft accidents and incident cases, which were input to Epaion to assess its effectiveness in diagnosing failures. The results of this evaluation, together with a brief description of Epaion, are presented. Additionally, in the present paper these results are compared with the results achieved by the rule-based process and the model-based process of DRAPHYS.

## Overview of Epaion

Epaion\* contains a self-organizing memory structured as a frame-based abstraction hierarchy, as defined by (Schank 1982), for storing previously encountered problems. Currently each case has been represented in a memory organization packet (MOP) as implemented in (Riesbeck & Schank 1989), but eventually MOPs will be implemented using LIMAP, a matrix-based knowledge representation tool (Feyock & Karamouzis 1993).

Each case is derived from an actual aircraft accident report and consists of a set of features that identify the particular accident, a list of the relevant context variables and

<sup>1</sup> Resident Associate of the National Research Council

<sup>2</sup> Work supported by NASA Grant NCC-1-159

\* Meaning "expert" in ancient Greek

their particular status, a set of observable symptoms, the fault, and a causal explanation that connects the observable symptoms to a justifying cause. The set of identifying features includes information such as aircraft type, airline, flight number, date of the accident, and similar data. The list of context variables includes information such as the phase of flight, weather, etc. The set of symptoms includes information about abnormal observations from mechanical sensors such as the value of the exhaust gas temperature, the value of engine pressure ratio, or from "human sensors", such as the sound of an explosion, or the smell of smoke in the passenger cabin. Cases containing all of this information are called *library cases*, whereas cases where the fault and the causal explanation are not available are called *input cases*. The following depicts a portion of an input case:

id:	Overseas National Airways
date:	November 12, 1975
phase-of-flight:	take-off
fuel-flow:	1 normal
	2 fluctuates
	3 fluctuates
	4 low
fault:	bird-ingestion
causal-event1:	ante bird-ingestion
	cnsq fan-blade-damage
causal-event2:	ante fan-blade-damage
	cnsq fan-rotor-imbalance

In the medical domain (Koton 1988) reports a system that is most closely related to ours. In contrast to this and most other CBR research efforts, each case in our methodology consists not only of a set of previously observed symptoms, but also represents sequences of events over certain time intervals. The time intervals are of unknown and uneven length; it is the event ordering that it is of importance. Such temporal information is necessary when reasoning about operating physical systems, since the set of symptoms observed at a particular time may represent improvement or deterioration from a previous reading, or may reveal valuable fault propagation information. In a jet engine, for example, the fact that the fan rotational speed was observed to be abnormal prior to an abnormal observation of the compressor rotational speed is indicative that the faulty component is the fan and that the fault propagated to the compressor, rather than the reverse.

In addition, the system incorporates a model called the *world knowledge model*. This model consists of two submodels: a functional dependency submodel with deep domain information about the functional dependencies between the components of the physical system, and a sub-

model representing causal information concerning transitions between various states of the physical system.

The functional dependency submodel is a digraph model of an aircraft system, with nodes representing primitive components, and "arrows" connecting nodes representing functional dependencies. Component B is said to be *functionally dependent* on component A if the proper functioning of B depends on the proper functioning of A. For example, the control surfaces of an aircraft are functionally dependent on the hydraulic system, since they will cease operating if the latter fails. The functional dependency submodel contains two kind of arrows, representing immediate and non-immediate links between components. Two components  $C_1$  and  $C_2$  are connected via an immediate link (I-link) when abnormal function of  $C_1$  at time  $t_1$  results in abnormal function of  $C_2$  at time  $t_2$  and (always)  $t_1 = t_2$ . If  $t_2 \geq t_1$  then  $C_1$  is said to be connected to  $C_2$  via a non-immediate link (N-link). For example, the engine driven pump (EDP) bypass valve is connected via an N-link to the EDP filter, but the EDP filter is connected to EDP bypass valve via an I-link.

The causality submodel of the world model contains information such as "fan-blade separation causes the rotational speed of the fan to fluctuate" and "the rotational speed of the fan causes the engine pressure ratio to fluctuate." Along with the causal information between two states, e.g. "inefficient air flow" and "slowing down of the engine", the model maintains a frequency count of the number of times that the system witnessed that inefficient air flow caused the engine to slow down.

Epaion's input constitutes a set of symptoms experienced by an airplane's crew during a flight. When the system experiences a new set of symptoms, i.e., when faced with an input (new) case, it searches its case library for the most similar case. This is done by placing the input case in self-organizing MOP memory under the most appropriate parents, determined as described in (Riesbeck & Schank 1989). The siblings may therefore be assumed to be closely related. The nearest sibling is retrieved as the case that is most on-point with respect to the input case.

When the system finds and retrieves a similar case, Epaion assumes that the current fault is the same as the fault in the retrieved case and adapts the causal explanation of the retrieved case to fit the current case. The fault and the causal explanation are both stored in the case library for future usage. The system is provided with a set of adaptation rules which, in addition to adapting the retrieved causal explanation to fit the current case, find possible gaps in the causal explanation and fill in the missing causalities by using the model. This causal explanation connects the symptoms to a justifying cause, and thus the system's causal reasoning ability produces a

causal analysis of the new case, rather than simply a reference to a previous solution. The new causal analysis is not only stored in the case library as part of the input case, but is used to augment and modify the causality knowledge of the world model.

At present, Epaion is implemented to diagnose faults in the engine subsystem of a generic twin engine transport. The programs currently run on various platforms using Common LISP.

### Epaion's Evaluation Approach

This section describes the experimental evaluation of Epaion on actual aircraft accident/incident cases involving engine faults. Information provided in the individual accident/incident reports from the National Transportation Board (NTSB), the British Air Accidents Investigation Branch (AAIB), and data collected from test accidents staged at Boeing Inc. (Shontz et. al. 1992) was used to derive the appropriate information constituting each case, a process called accident reconstruction. We reconstructed a total of eighteen cases, of which twelve were used as library cases, and six as input cases.

Accident reconstruction is not a straightforward process and has its limitations. In the reconstruction process the symptoms from all accidents had to be identified from the sources that described the accidents. Unfortunately, numerical sensor data from the engine parameters was not available, so the symptoms were used as reported in (Shontz et. al., 1992), or derived based on the descriptions in the NTSB or AAIB analysis of each accident. NTSB and AAIB reports did not always explicitly describe the symptoms in each case; even in those cases where symptoms were mentioned explicitly they were usually only those described by the flight crew. The sequence of symptoms could therefore not always be determined completely.

In addition a chain of causalities had to be constructed for each of the accidents used as library cases. This chain explains each observed symptom by connecting the symptom to a justifying cause. Determining the causal explanation of the symptoms for each case was a difficult task because of a paucity of definitive experts who could provide this information. While pilots, maintenance personnel, and aircraft system designers are all knowledgeable about some aspects of aircraft diagnosis, each has deficiencies in one area or another. The causal explanations used in each library case were constructed after interviewing personnel with expertise in the above fields, and consulting NTSB and AAIB reports.

The evaluation process required that each input case be presented to Epaion separately, and that the system produce a diagnosis along with a causal explanation. The diagnosis produced by Epaion was then compared with the correct diagnosis for the particular scenario. In addition, the reasoner was evaluated based on the number of symptoms for which the reasoner was able to find a justification. A "correct diagnosis" is the diagnosis determined by NTSB, AAIB, or by (Shontz et. al. 1992). Epaion is said to have produced a *complete explanation* if the system was able to explain each observed symptom by connecting the symptom to a justifying cause.

Table 1 presents a summary of the results. The first two columns identify each scenario that was presented to Epaion as an input case. The following two columns identify the appropriate classification of the accident/incident along with the actual fault as determined by either the NTSB, the British Air Investigations Branch, or Boeing's test data. The fifth and sixth columns present the classification of each accident/incident done by Epaion along with the fault assumed by Epaion. The last column tabulates the result of Epaion's adaptation phase. Epaion's explanatory performance was characterized as complete in the cases where the system was able to causally justify every symptom experienced in the input case.

	Case	Correct		Epaion's		
	Identification	Classification	Fault	Classification	Fault	Explanation
1	G-OBMG	Rotor Scenario	Fan Blade	Rotor Scenario	Fan Blade	Complete
2	American Airlines 566	Rotor Scenario	Turbine Blade	Rotor Scenario	Fan Blade	Complete
3	China Air 006	Fuel Scenario	Fuel Controller	Fuel Scenario	Fuel Subsystem	Complete
4	Galunggung	Volcanic Scenario	Volcanic Ingestion	Volcanic Scenario	Volcanic Ingestion	Complete
5	Southern Airways 242	Water Scenario	Water Ingestion	Miscellaneous Scenario	Foreign Object Ingestion	Complete
6	Boeing Test Flight F5	Icing Scenario	Ice Ingestion	Icing Scenario	Ice Ingestion	Incomplete

Table 1: Summary of Epaion's Results

## A Rule-Based/Model-Based Approach

In this section Epaion's method is compared to a rule-based/model-based approach, as exemplified by DRAPHYS.

DRAPHYS\* operates in the same domain as Epaion. It consists of by two discrete subsystems; a rule-based reasoner and a model-based reasoner. DRAPHYS' rule-based reasoner involves compiled knowledge about the association between symptoms and faults. Its output is a set of fault hypotheses.

DRAPHYS' model-based reasoner operates as follows.

Case Identification	Actual Fault	Rule-Based Reasoner	Model-Based Reasoner
1 United Flight 611	Turbine Blade	Turbine Blade	Turbine Blade
2 National Flight 27	Fan Failure	Turbine Blade	Fan Failure
3 Northwest Flight 79	Fan Failure	Turbine Blade	Fan Failure
4 Overseas National Flight 32	Foreign Object Ingestion	(none)	(not complete)
5 Southern Airways 242	Water Ingestion	Flameout	(incorrect set)
6 American Flight 191	Engine Separation	Flameout	Engine - Fan
7 Air Florida Flight 2198	Turbine Disk	Turbine Blade	Combustor, Turbine, EPR
8 Eastern Airlines Flight 935	Bearing Failure	Flameout, Turbine Blade	Compressor

**Table 2:** Summary of DRAPHYS' Results

First, based on the observed symptoms, the fault is localized to a particular subsystem in the aircraft. Each component in that particular subsystem is suspected as the source of the fault. Then for each suspected component the reasoner uses simulation to determine the fault propagation behavior. The simulation is done with the help of the same functional and physical dependencies models as used in Epaion. The fault propagation behavior produced by the simulation is then compared with the current status of the components. If this fault propagation involves components which currently are not affected, then the suspected faulty component no longer constitutes a possible source of the fault. The output of this reasoning process is a list of fault hypotheses which hopefully is smaller than the entire set of components in a particular subsystem of the aircraft.

An empirical evaluation of DRAPHYS was performed by presenting eight aircraft accident/incident cases which were reconstructed from actual NTSB reports (Abbott 1991; Schutte). Table 2 presents a summary of the results.

For the first case both reasoners in DRAPHYS produced the correct diagnosis. For the second and third cases the

model-based reasoner produced the correct diagnosis. The rule-based reasoner contains no rules that can identify a fan failure, thus due to the similarities in the symptoms for turbine blade separation and fan failure the rule-based reasoner misdiagnosed the fault as a turbine blade separation.

Although there is a rule that may identify foreign object ingestion, the rule-based reasoner failed to produce any diagnosis for the fourth case. For the same case the model-based reasoner assumed that all the major components in the engine were valid fault hypotheses. The failure to trim the set of suspected components is due to the fact that symptoms occurred on all major engine sensors si-

multaneously.

In the absence of rules that could identify massive water ingestion and engine separation for the fifth and sixth case the rule-based reasoner diagnosed a flameout. This may be considered as the correct diagnosis, since a flameout is a direct consequence of both faults. The model-based reasoner produced an incorrect set of hypotheses for the fifth case. For the sixth case it hypothesized a physical propagation from an engine failure to the hydraulic subsystem. Worth mentioning here is that the model-based reasoner recognized that the abnormal sensor readings from the hydraulic system sensors were not the result of a failure in the hydraulic system, but a physical propagation from an engine failure.

For the seventh case the rule-based reasoner assumed a turbine blade as being the responsible faulty component. This diagnosis may be considered correct since all the blades and the disk were separated. For the same case the reason that the model-reasoner could not identify the turbine as the sole responsible component is that there is no specific sensor monitoring the turbine. For the last case an incomplete rulebase resulted in an incorrect diagnosis by the rule-based reasoner, while incompleteness in the physical dependencies model - no physical link in the

\* Diagnostic Reasoning About Physical Systems

model between the compressor and the hydraulic line - lead to misdiagnosis by the model-based reasoner.

## Conclusions

This paper gives a brief description of two different reasoning approaches in the domain of in-flight fault management of aviation subsystems, particularly jet engines. From the empirical evaluation of these approaches it is evident that despite the minor shortcoming of each approach automation of inflight diagnosis and prognosis as an aid to the flight crew has great potential for improving the general safety of civil transport operations.

Generally we see that when the rule-based approach failed to produce the correct diagnosis this was mainly due to the absence of specific rules that could identify certain faults. The incompleteness of the rule-base was the reflection of the difficulty for knowledge elicitation in this domain rather than the product of a "sloppy" implementation. On the other hand the inability to model and monitor the world at a great level of detail degraded the performance of the model-based reasoner.

The case-based reasoner achieved its results by not relying exclusively on the models. Additionally, inherited by the nature of the case-based methodology, Epaion reduced the problem of knowledge elicitation to mere description of past experiences. Based on the results of Epaion's empirical evaluation we make the following observations:

a. An expanded case library will enhance the systems classification capability and therefore offer better matches for each additional input case.

b. Presenting the system with cases which are reconstructed based on an accurate set of symptoms is vital for correct matching and therefore correct diagnoses.

c. The more knowledge that the system contains in its abstraction hierarchy, the better its explanation capability will be.

A major concern of this project has been to create a system capable of achieving a practically useful level of performance on a case base of significant size, thereby avoiding the "toy problem" trap besetting many AI systems. The extensive use of a classification hierarchy allows the system to achieve  $O(\log n)$  search times, while the information abstraction attendant on accident reconstruction produces space-efficient representations. The system is currently hosted on a desktop personal computer, and is estimated to be capable of storing the full set of propulsion related aircraft accident for the last 20 years. These considerations, together with the encouraging level of success achieved by Epaion, support the expectation that this system will prove to be an effective contributor to aircraft safety.

## References

- Abbott, K. H. (1991). Robust Fault Diagnosis of Physical Systems in Operation, NASA Technical Memorandum 102767, NASA Langley, Hampton, Virginia.
- Feyock, S., and Karamouzis, S. (1993). LIMAP. NASA Contractor Report 4539, NASA Langley Research Center, Hampton, Virginia.
- Karamouzis, S., and Feyock, S. (1993a). "A Case-Based Approach to Handling Aircraft Malfunctions", Proceedings of SPIE's Applications of AI XI: Knowledge-Based Systems in Aerospace & Industry, Orlando, Florida, April 1993.
- Karamouzis, S., and Feyock, S. (1993b). "A Performance Assessment of a Case-Based Diagnostic System for Aircraft Malfunctions", Proceedings of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, City Chambers, Edinburgh, June 1993.
- Koton, P. (1988). Using Experience in Learning and Problem Solving. Ph.D. Thesis MIT/LCS/TR-441. MIT Press, Cambridge, Massachusetts.
- NTSB-AAR-72-9. Aircraft Accident report: United Airlines, Inc., Boeing 737-222, N9005U, Philadelphia International Airport, July, 19, 1970, National Transportation Safety Board, NTSB-AAR-72-9.
- Riesbeck, C. K., and Schank, R. C. (1989). Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Hillsdale, New Jersey.
- Schank, R. C., (1982). Dynamic Memory: A Theory of Learning in Computers and People. Cambridge University Press.
- Schutte, P. C., Abbott, K. H., & Ricks, W. R. A Performance Assessment of a Real-time Diagnostic System for Aircraft Applications, Unpublished Technical Report, NASA Langley Research Center, Hampton, Virginia.
- Shontz, W. D., Records, R. M., and Antonelli, D. R. (1992). "Flight Deck Engine Advisor Final Report," NASA Contractor Report 189562, Langley Research Center, Hampton, Virginia.