The Use of Multiple Models in Case-Based Diagnosis+

Stamos T. Karamouzis*   Stefan Feyock**

Department of Computer Science
College of William & Mary
Williamsburg, VA 23185

Abstract

The work described in this paper has as its goal the integration of a number of reasoning techniques into a unified intelligent information system that will aid flight crews with malfunction diagnosis and prognostication. One of these approaches involves using the extensive archive of information contained in aircraft accident reports along with various models of the aircraft as the basis for case-based reasoning about malfunctions.

Case-based reasoning draws conclusions on the basis of similarities between the present situation and prior experience. We maintain that the ability of a CBR program to reason about physical systems is significantly enchanced by the addition to the CBR program of various models. This paper describes the diagnostic concepts implemented in a prototypical case-based reasoner that operates in the domain of in-flight fault diagnosis, the various models used in conjuction with the reasoner's CBR component, and results from a preliminary evaluation.

Introduction

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 changes dynamically by modifying its set of components, the components' states and pattern of interconnections, and the system's behavior.

To address these concerns a prototypical case-based reasoner (CBR), called Epaion, has been developed by the Intelligent Cockpit Aids Team at NASA Langley Research Center, in connection with ongoing work on AI-based systems for in-flight fault management [Schutte et al.]. The reasoner operates in 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.

Several aspects of the aircraft domain make automation of in-flight diagnosis challenging. In contrast with non-operative diagnosis (i.e., diagnosis of systems that can be shut down), symptoms in aircraft subsystems may change with time because of failure propagation. Information about the operational status of many aircraft components may be unavailable or incomplete due to limited instrumentation, and safety and

comfort considerations place further constraints on in-flight testing.

The approach we are taking employs a novel methodology for dealing with physical systems in operation, and involves the use of case-based techniques in conjunction with models that describe the physical system. Case-Based Reasoning systems solve new problems by finding solved problems similar to the current problem and adapting their solutions to the current problem, taking into consideration any differences between the current and previously solved situations. Because CBR systems associate features of a problem with a previously derived solution to that problem, they are classified as associational reasoning systems.

We maintain that the ability of a CBR program to reason about physical systems can be significantly enchanced by the addition of various models to the CBR program. This paper describes the diagnostic concepts implemented in Epaion[1], the various models used in conjuction with the CBR component, and results from Epaion's preliminary evaluation. Although the examples presented pertain to aircraft malfunctions, it is clear that these techniques are applicable to spacecraft as well.

Knowledge Sources

Epaion draws its power from several knowledge sources, including a library of aircraft accident/incidents; a functional dependency model with deep domain information about the functional dependencies between the components of the aircraft; and a model representing causal information concerning transitions between various states of the aircraft.

---

[1] Ancient Greek for "expert"

Case Library

Epaion maintains a library of actual aircraft accident/incident scenarios called *cases*. Each case consists of a set of features that identify the particular scenario, a list of the relevant context variables and 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, the 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*.

In contrast to 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 may have unknown and unequal lengths; it is the event ordering that 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.

Causality Model

Epaion's causality 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.

Functional Dependency Model

The functional dependency model 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 model 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 C_1's failure propagates immediately to C_2, i.e., abnormal function of C_1 at time t_1 results in abnormal function of C_2 at time t_2 and $t_1 = t_2$. If $t_2 > t_1$ then C_1 is said to be connected to C_2 via an non-immediate link (N-link). For example, if the fan belt in an automotive engine breaks, the fault propagates immediately to the electrical system, as indicated by the generator light, but it may take some time before the propagation to the cooling system becomes evident from the temperature sensor.

Physical Dependency Model

The physical dependency model is a digraph of an aircraft system, similar to the functional dependencies diagraph, in which the links in the graph represent potential paths of fault propagation due to physical proximity. This sort of propagation occurs when uncontrolled discharges of energy attendant on component malfunctions propagate to neighboring systems. The severing of nearby hydraulic lines by blade fragments from a disintegrating turbine provides a typical example.

The Abstraction Hierarchy

The Case-Based Reasoning component of Epaion consists of a self-organizing memory structured as a frame-based abstraction hierarchy, as defined by [Schank 1982]. This memory forms an upper bounded semi-lattice that contains domain specific information at different levels of abstraction. The information contained in the lattice includes:

a. The names of all components in an aircraft engine.

b. The components that are sensors. The exhaust gas temperature, the rotational speed of the fan, and the fuel flow indicator are some of the mechanical sensors in an aircraft's engine. Vision, sight, and smell are the "human sensors" used in the diagnostic process.

c. The possible values for each sensor. For a mechanical sensor the allowable values are: lower than expected; normal; higher than expected. If a sensor initially indicates values that are normal, then at the following time interval indicates values that are lower than expected, and at the third time interval still indicates values which are lower than expected, then the status of the sensor during these three time intervals is *normal, lower, lower* which is a kind (i.e., subcategory) of *overall lower* than expected status which in turn is a kind of *abnormal* status.

d. The various faults that may be observed in an engine subsystem. For example, it is represented that *seagull ingestion* is a kind of *bird ingestion* fault which is a kind of *foreign object ingestion* fault and so on.

e. Information on how faults manifest themselves. For example, fan vibration and abnormality in the rotational speed of the fan are manifestations of a problem in the fan.

f. The accident/incidents that the system already knows. For example the system knows that the incident of a China Airlines Boeing 747 that suffered a mishap over the Pacific Ocean on February 19, 1985 [NTSB-AAR-86-03] is an instance of an accident/incident since it is a kind of *rotor related scenario* which is a kind of *engine related scenario* which is a kind of *accident/incident scenario*.

Reasoning Cycle

Epaion's reasoning cycle consists of the following three phases: input a new problem; retrieve the most similar case; adapt the retrieved case to fit the current scenario.

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[2] 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 similar to the input case.

When the system finds and retrieves a similar case, Epaion assumes that the current fault is the

---
2 Memory Organization Packet

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 models. This causal explanation connects the symptoms to a justifying cause, and thus the system's multiple-model-based 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 knowledge of the causal model. The following section provides details of this process.

Adaptation and the Models

Epaion's adaptation algorithm is summarized in the following two steps:

The first step involves the transfer of the fault from the library case in the input case and consists of two possibilities.

Case 1: If the match between the input case and the library case exceeds a threshold value then the fault is transferred intact. For example, if in the library case the fault was a malfunctioning fuel controller, then it is assumed to be the same in the input case.

Case 2: If the match is below the threshold value then an abstraction of the library case fault is transferred to the input case. For example, if in the library case the fault was bird ingestion, then it is assumed that in the input case the fault is foreign object ingestion.

The second step involves the adaptation of the causal explanation of the library case so it can explain each, or as many as possible, of the

symptoms of the input scenario by connecting them to a justifying cause. This consists of the following possibilities:

Case 1: If the library case and the input case have identical symptoms then the causal explanation of the library case is transferred intact to the input case.

Case 2: If the input case contains symptoms that do not appear in the library case then the causal explanation of the library case is transferred in the input case and the following additional processing takes place. Let $\phi2$ be an unexplained input case symptom.

Subcase 1: If the causal model contains the relation $\phi1$ *causes* $\phi2$, and $\phi1$ is a symptom or manifestation in the input case, then the link $\phi1$ *causes* $\phi2$ is added in the causal explanation of the input case.

Subcase 2: The causal portion of the model does not contain the relation $\phi1$ *causes* $\phi2$, but the functional dependency model knows that component C_2 is functionally dependent on component C_1, and $\phi1$ is a manifestation of abnormal behavior of component C_1, and similarly $\phi2$ is a manifestation of C_2. This knowledge is depicted by the graph
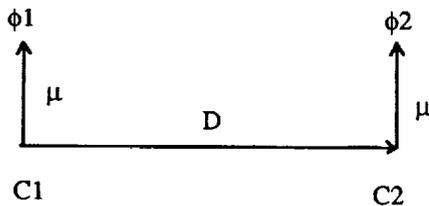
where ϕ denotes a phenomenon that is a symptom or manifestation μ of abnormal behavior of a component. Additionally, if $\phi1$ is a symptom in the input case and time($\phi1$) $\leq$ time($\phi2$), i.e., symptom $\phi1$ appeared before or concurrent with $\phi2$ then the link $\phi1$ *causes* $\phi2$ is added in the causal explanation of the input case.

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. Figure 1 displays the use of the various models during the adaptation process.
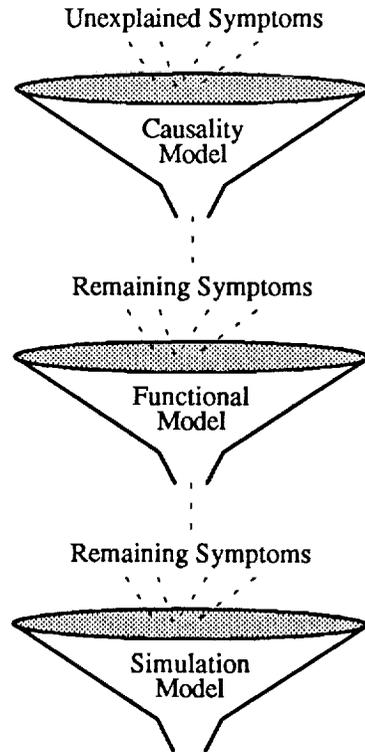

Figure 1: Use of models during adaptation

Simulation and the Physical Model

We have indicated that Epaoin uses a physical dependency digraph as one of its models. This is a makeshift measure, however, due to the fact that physical fault propagation, being the result of catastrophic component failures, is highly unpredictable. One expedient for dealing with this unpredicatability is to refer to previous cases, as Epaion does; another is to utilize spatial simulation models (SSMs) to determine the effect of uncontrolled energy releases. [Feyock & Li, 1990, 1992] describe the use of SSMs to

predict both fluidic and energy leaks[3]. These models, which are easily interfaced with host systems, require only the identity of the faulty component, which can be supplied by Epaion. The SSM then looks in the component database to determine the location and type of the component. If the component is of a type that can cause a fluid or energy leak, the system uses this information to set the initial conditions for the simulation. The simulation is then run, and the physical propagation paths predicted by the SSM are extracted from the run data.

In addition to addressing the chaotic nature of physical propagation, our use of simulation models in conjunction with more traditional reasoning systems is prompted by a belief that deriving answers to real-world questions by setting up the initial conditions of simulation models, running the simulations, and extracting information from the results of the run, constitutes a powerful but underutilized mode of operation for AI systems.

Results

We conducted an 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 sixteen were used as library cases, and six as input cases.

The evaluation process required that each input case be presented to Epaion separately, and that

[3] We denote as "energy leaks" the catastrophic disintegration of components due to the uncontrolled release of kinetic or potential energy.

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. The results achieved are very promising for the future success of the system. Based on the results we make the following observations.

- Classification

Five of the six cases in this evaluation were correctly classified. A case involving water ingestion [NTSB-AAR-78-3] was classified under the category of miscellaneous scenarios due to the lack of previously encountered water ingestion scenarios. An, expanded case library will enhance the systems classification capability and therefore offer better matches for each additional input case.

- Diagnosis

Epaion was able to correctly diagnose five of the six scenarios. A case representing the American Airlines Flight 566 scenario [NTSB-F-A067] was properly classified as rotor scenario but misdiagnosed as fan problem rather than turbine problem. This is a result of the fact that problems in the fan and problems in the turbine manifest themselves similarly, and therefore both kinds of faults are classified under the category of *rotor scenarios*. When the American Airlines scenario was used as input case the system retrieved as the most similar case a Dan Air incident [AAI-AAR-4/90], which is a fan blade scenario. With almost negligible difference in the degree of match between the input case and the

C-2

relevant library cases, the second best match was the accident of the United Airlines Flight 611 that took place on July 19, 1970 [NTSB-AAR-72-9]. This is a turbine fault scenario and would have achieved a higher degree of similarity with the input case if the time order of the symptoms in both cases had been represented more precisely. All symptoms used in reconstructing the case of the United Airlines Flight 611 were based on expert opinion, but none were explicitly stated in the NTSB report. With the exception of the behavior of the EGT, the same holds for the symptoms used to reconstruct the American Airlines Flight 566 scenario. This suggests that presenting the system with cases that are reconstructed based on an accurate set of symptoms is vital for correct matching and therefore correct diagnoses.

• Symptom explanation

In five of the cases presented as input Epaion was able to explain all of the symptoms experienced. When Epaion was presented with the symptoms of an icing scenario staged at Boeing [Shontz et. al. 1992] it failed to explain the presence of broad-band vibration. The failure is attributable to insufficient information in the abstraction hierarchy. If the fact that broad-band vibration is a manifestation of fan abnormality had been included in the abstraction hierarchy, the system's functional dependencies model would have explained the broad-band vibration symptom as a result of fan blade damage. The same result would have been achieved if the system had previously experienced other cases with broad-band vibration, thus enabling the causal model to explain the vibration. It is evident that the more knowledge the system contains in its abstraction hierarchy, the better its explanation capability will be. Current efforts are accordingly focused on expanding this knowledge to a substantial size.

Conclusion

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. The Epaion Case-Based Reasoning system we have developed for the purpose of performing fault diagnosis and prognosis of aircraft in operation uses a hybrid reasoning process based on a library of previous cases and several types of models of the aircraft as the basis for the reasoning process. This arrangement provides the methodology with the flexibility and power of first-principle reasoners, coupled with the speed of associational systems.

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 with 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

AAI-AAR-4/90. (1990). *Report on the accident to Boeing 737-400 G-OBME near Kegworth, Leicestershire on 8 January 1989*, Air Accidents Investigation Branch, AAIB-AAR-4/90.

Feyock, S., and D. Li, Simulation-based Reasoning about the Physical Propagation of Fault Effects, 1990 Goddard Conference oh Space Applications of Artificial Intelligence, NASA/Goddard Space Flight Center, Greenbelt, Md, May 1990

Feyock, S., and D. Li, Qualitative Spatial Reasoning about Fault Effect Propagation, Computer Science Department Technical Report, College of William & Mary, Williamsburg, Va, 1992.

Feyock, S., and S. Karamouzis (1991). LIMAP . Technical Report WM-92-1, College of William & Mary, Computer Science Department, Williamsburg, Virginia.

Karamouzis, S., & S. Feyock (1992). "An Integration of Case-Based and Model-Based Reasoning and its Application to Physical System Faults", *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Lecture Notes in Artificial Intelligence, No. 604 (F. Belli and F. J. Radermacher, Eds), Springer-Verlag, Berlin, 100 - 108, 1992.

Lloyd, A. T. (1990). "Vulcan's Blast", *Airliner*, April-June 1990.

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.

NTSB-AAR-76-19. *Aircraft Accident report: Overseas national Airways, Inc., DC-10-30, N1032F, John F. Kennedy International Airport, Jamaica, new York, November 12, 1975*, National Transportation Safety Board, NTSB-AAR-76-19.

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

NTSB-AAR-86-03. *Aircraft Accident report: China Airlines, Boeing 747-SP, N4522V, 300 Nautical Miles Northwest of San Francisco, California, February 19, 1985*, National Transportation Safety Board, NTSB-AAR-86-9.

NTSB-F-A067. *Aircraft Accident report: American Airlines, Inc., Boeing 727-023, Erlanger, Kentucky, May 21, 1978*, National Transportation Safety Board, File No. IAD-78-F-A067.

Riesbeck, C. K., and R. C. Schank (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., K. H. Abbott, and W. R. 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., R. M. Records, and D. R. Antonelli (1992). ``Flight Deck Engine Advisor Final Report," NASA Contractor Report 189562, Langley Research Center, Hampton, Virginia.