# An Evolvable Tri-Reasoner IVHM[1] System

In Alphabetic Order: Les Atlas (University of Washington), George Bloor (Boeing[2]), Tom Brotherton (IAC), Larry Howard (Institute for Software Integrated Systems), Link Jaw (Scientific Monitoring Inc.), Greg KACPRZYNSKI (Impact Technologies), Gabor Karsai (Institute for Software Integrated Systems), Ryan Mackey (Jet Propulsion Laboratories), Jay Mesick (Boeing), Rick Reuter (Boeing), Mike Roemer (Impact Technologies)

*Abstract*—Meeting the challenges of decreasing operational costs and increasing operational readiness for future aircraft will require a systemic approach to integrated vehicle health management (IVHM). Realizing such an approach will involve synergistic deployments of component health monitoring technologies, as well as integrated, model-based reasoning capabilities for the interpretation of these monitors' outputs. Further, it will involve the introduction of learning technologies to support the continuous improvement of the knowledge enabling these reasoning

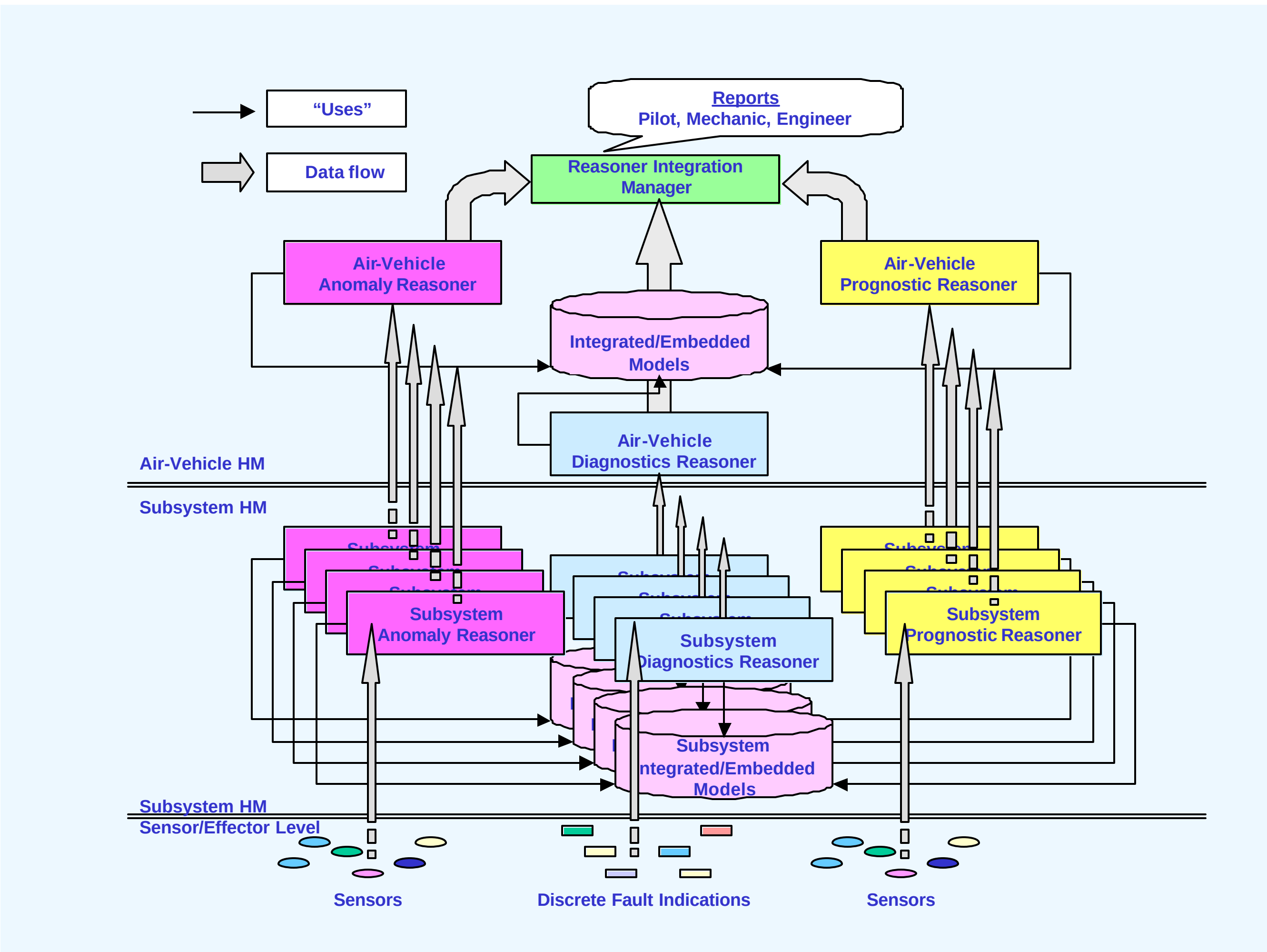



**Figure 0 – The Tri-Reasoner Integrated Vehicle Health Management System**

[1] Integrated Vehicle Health Management (IVHM)

[2] Copyright ©1999, The Boeing Company, all rights reserved.

capabilities. Finally, it will involve organizing these elements into an architecture that governs integration and interoperation-within the VHM system, between its on-board elements and their ground-based support functions, and between the VHM system and external maintenance and operations functions. In this paper we present and discuss architecture for an evolvable tri-reasoner integrated VHM system, its particular elements and their interrelationships.

TABLE OF CONTENTS



## 1. INTRODUCTION

The purpose of the on-board health management system is to alert those who need knowledge of the aircrafts health. Similarly, the purpose of an evolvable system for constructing the on-board health management software is to monitor and manage the health of the on-board maintenance software, itself. One of the constant challenges for commercial and military aircraft is to build on-board maintenance software that is effective and provides reliable information. Experience finds us with maintenance systems, themselves, in need of maintenance, and this is cost prohibitive. *It must be assumed that we are not able to design the maintenance systems correctly the first time.* As technology upgrades proceed in time, these changes in the air-vehicles architecture further confound the ability of the maintenance software to keep pace. The challenge is to design a software development framework that accommodates rapid and *evolvable* upgrades to the maintenance software.

| ACRONYM | MEANING |
|---|---|
| VHM | Vehicle Health Management |
| IVHM | Integrated VHM |
| IM | Integrated Model |
| LRU | Line Replaceable Unit |
| SNR | Signal to Noise Ratio |
| HM | Health Management |
| A/D/P | Anomaly/Diagnostics/Prognostics |
| AR | Anomaly Reasoner |
| AVAR | Air-Vehicle Anomaly Reasoner |
| DR | Diagnostic Reasoner |
| AVDR | Air-Vehicle Diagnostic Reasoner |
| MTTF | Mean Time To Failure |
| RIM | Reasoner Integration Manager |
| API | Application Interface |
| FM | Failure Mode |
| BITE | Built In Test Equipment |
| LED | Light Emitting Diode |

*RIM Output Integrity*

The tri-reasoner integrated vehicle health management conceptual (IVHM) framework, depicted in Figure 1, is composed of a reasoner integration manager (RIM) (green box), and three independent views of the vehicle's health. These views are created through the use of three system reasoners (anomaly, prognostic, and diagnostic) whose algorithms traverse the *integrated model*[3]. Health management reports are output from the RIM and their integrity depends upon the integrity of the tri-reasoner algorithms, the data they process and the integrity of the integrated model (IM), Figure 8. The tri-reasoner algorithms are generic and decoupled from any domain knowledge to enable the use of algorithms that have withstood a wide variety of applications thus increasing their integrity. The domain knowledge is captured in the integrated model (IM). It is paramount that the process used to capture this knowledge in the IM has high integrity.

Fundamental to portraying the effects of failures accurately in the integrated model is a cross validation by the design engineers. Typically engineers use their domains favorite software tool to model and design their own niche of the aircraft. The task of maintenance software design involves all aspects of the aircraft; it is the penultimate cross-disciplinary task. Aspects of the models that these design engineers' use are re-usable within the integrated model. How does the maintenance software engineer populate the integrated model from models created by the primary functional design engineers? To enable this, a tool integration framework was developed, (see Figure 2). [1,2,3]. With this tool integration framework in place, as the designers' upgrade their models, relevant aspects of these models flow into the integrated model; thus helping to evolve the on-board maintenance software and provide an integrity check simultaneously

Finally, there is an untapped source of information that occurs naturally in modern complex commercial and military aircraft designs. A plethora of data and signals that are used for primary aircraft functionality are available for other purposes. The anomaly detection and reasoning system is designed to take advantage of the availability of this information to listening in on the aircraft health, like a stethoscope; to try to find anything unusual and related to events observed in the prognostic and diagnostic systems. It

[3] To be discussed below.

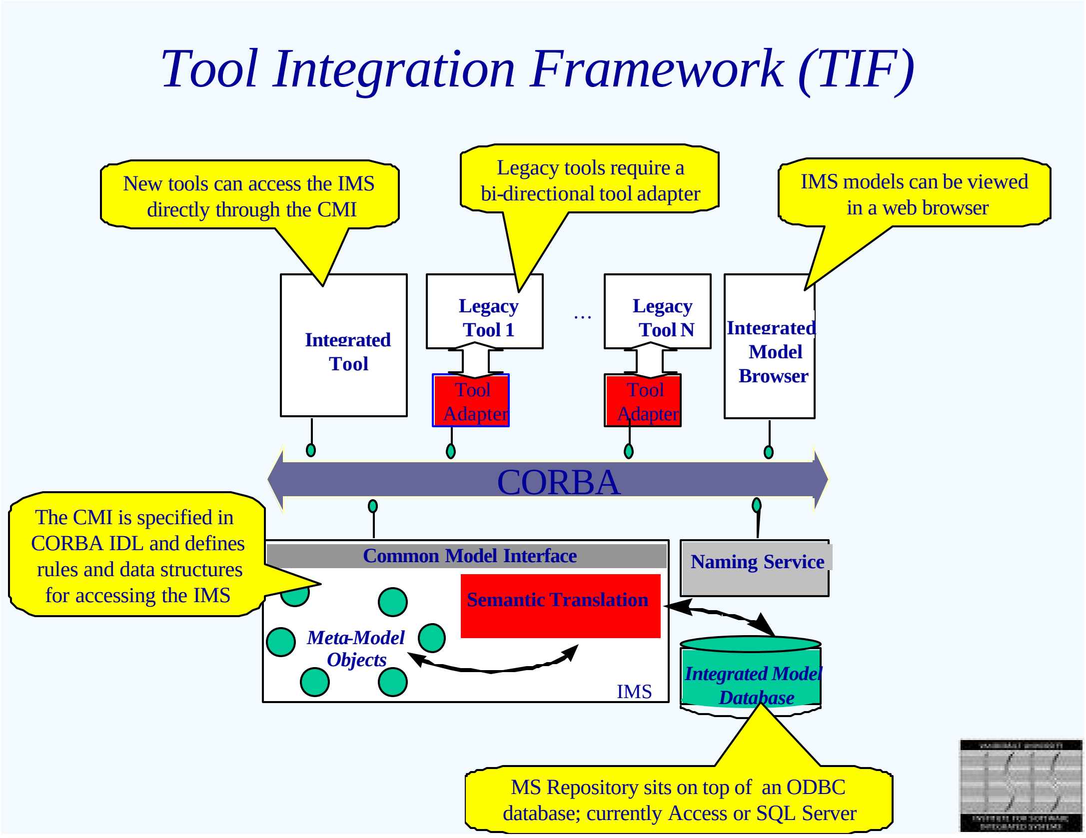



**Figure 2 - A Tool Integration Framework to Populate the Integrated Model**

is a fundamental component of our tri-reasoner IVHM system.

This paper discusses the forgoing topics including: 1) the anomaly detection and reasoning system. 2) the prognostic reasoning system, 3) the diagnostic reasoning system, 4) the integrated model, and 5) the reasoner integration manager.

## 2. BACKGROUND

*Anomaly Detection – Incipient Failures, Intermittent Failures, Active Failures, Novel Events, and Prognostics*

Any given air-vehicle has its unique characteristics due to unavoidable sources of variability. These sources include manufacturing, both across and within manufacturers. Variability appears in the onboard systems (e.g. mechanical, electrical, and hydraulic). An aircraft's reaction to ambient environmental conditions, such as altitude, depends upon its age and it varies across the fleet. Part replacement and repair modifies these unique characteristics, as well. In the presence of this variability, on-board health management systems are challenged to manage incipient and intermittent failures as well as active faults. Acknowledging these copious sources of variability, we now provide an overview the following system behaviors: off-nominal, incipient failure, intermittent failure, active failures, an anomalous event, and a novel event. An overview of the challenges of creating prognostic algorithms and reasoners is also provided.

We define the nominal behavior of a given air-vehicle to be that behavior that exists when all intended functionality is available and operating within the constraints of the intended design *at a given point in time (or an averaged window of time).* Note, the same aircraft can be considered functioning, as intended, at two different points in time even though the characteristics of individual component and sensor characteristics may have changed.

Suppose we measure and characterize the aircraft at time intervals $t_1$ and $t_2$. Assume the aircraft is performing within specification and as desired from every possible perspective. Furthermore suppose there is a difference in the measured parameters even though ideally we would like them to be the same. Then we define the measurements at time interval $t_1$ to characterize the nominal performance of the aircraft. And, at time interval $t_2$ we construct a distance metric between the measured parameters at these two time intervals. We are now in a position to characterize the baseline behavior of the air-vehicle. This is the initial step towards: 1) anticipating future behavior and 2) providing a context for understanding current and undesirable behavior. The mechanism for characterizing baseline performance and identifying deviations from the baseline is defined as *anomaly detection.*

If the observed behavior of the air-vehicle is deemed to have significantly departed from its baseline then we say that we have observed an anomalous event. An anomaly is any off-nominal behavior including any failure described as *incipient*, *intermittent*, or *active*. A failure event is a subcategory of an anomalous event.

An active failure is off-nominal behavior of the air-vehicle that also displays unintended functionality. Similarly, an intermittent fault is an active fault that does not persist.

An incipient failure is a system or component that is still operational, but is observed to be transitioning towards a failed condition.

Fault monitors within a physical piece of hardware exist to declare the health of the hardware to aid the maintainer. These monitors may be the victims of faulty signals being passed into them by upstream components. It is the role of the air-vehicle diagnostic system to construct the integrated perspective and isolate the fault source(s). The algorithms that perform these tasks are called diagnostic reasoners. Traditional diagnostic reasoners for air-vehicles rely on the health reports (discrete 1,0) emanating from the line replaceable units (LRU) as the primary source of information.

A fault monitor detects a failed condition when it occurs or shortly thereafter. An anomaly detector responds prior to a fault as well as during a failed condition. The anomaly detector is used primarily to identify novel events not seen by the myriad of fault monitors already implemented. This will be discussed in detail, later. An anomaly detector, like a prognostic algorithm, responds to incipient faults; it does not have the task of predicting when the fault will actually occur. A prognostic is the ability to assess the current health of a part and predict into the future its health for a fixed time horizon or predict the time to failure. The ability to perform reliable prognostics is the key to condition based maintenance (CBM). Prognostics are critical for improving safety, planning missions, scheduling maintenance, and reducing maintenance costs and down time.

There currently exist simple prognostics for component parts in the form of component *life monitors*. Life monitors are usually based on statistics gathered over a large population of components. Sometimes they do include physical models. However even these models are a measure of an average component's health and are not tailored to the specific component being monitored. Component life monitors are coarse and conservative. And all are essentially based on measuring "time" in some fashion (for example "cycles"). The ideal goal is to develop algorithms that make decisions from current measurements of a component to develop a *component-specific* prognostic.

*Problem Anatomy – 100% Healthy to 100% Failed*

Figure 3 shows the different component health monitoring problems that need to be addressed. This figure shows the trajectory of a machine component's health as a function of time. When the component is new, its health is considered 100 percent. As time goes on and the component begins to wear out, it's health, defined here somewhat arbitrarily, drops. This figure assumes the component is following a known fault life degradation path. In the discussion following, an *anomaly* is any off nominal operating condition. Anomalies come in two types. The first is a *fault*. A fault is a known off nominal condition. It is assumed that fault-specific algorithms have been developed to detect a fault. The second anomaly is a *novel* event. A novel event is an unknown off-nominal condition. That is, the novel event is not nominal nor is it classified in any of the known fault conditions. *It's something completely new*. We do not know if the novel event is an active failure, an incipient failure, or an "I don't care". Prognostic algorithms are designed to respond to "known faults" with know failure modes (and not novel events). This is because an important part of the prognostics is the modeling for prediction of the component health trajectory shown in Figure 3. In order to develop that model, something about the trajectory of a component from nominal to a known fault condition is required.

Component health monitoring determines where the component is on the curve shown in Figure 3. Is the part "nominal"? Does some "anomaly" condition exist? Or, is it some where between those two extremes? Note that a normal component health curve may encompass a variety of behaviors and thus this curve represents a single region or

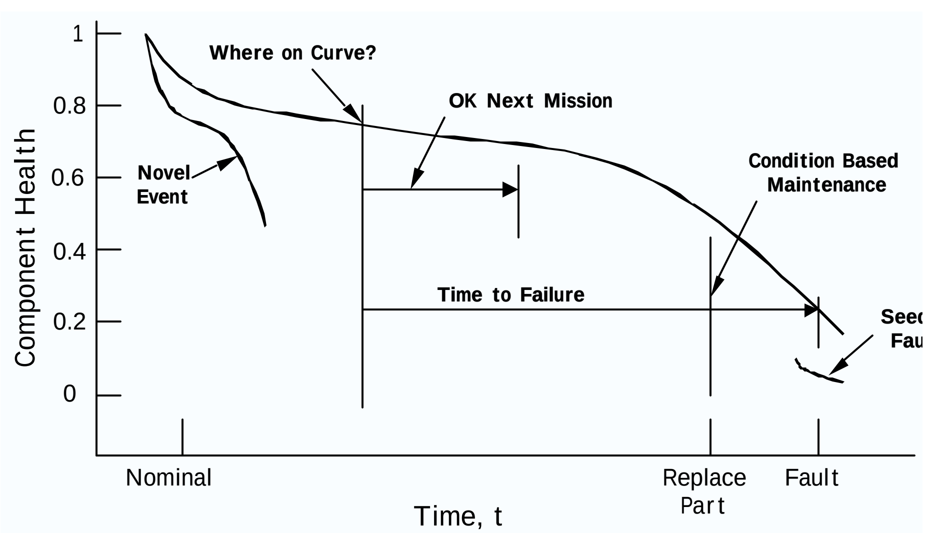



**Figure 3 - Component Life: 100% Healthy to 100% Failed**

single fault trajectory rather than a series of strictly defined points. Determining where we are on the component health curve is the first step in prognostics.

Fault detection / diagnostic reasoning as discussed above, determines if a component has moved away (degraded) from 100% along a known path, as indicated in Figure 3, to a point where component performance may be compromised. Novelty detection determines if the component has moved away from what is considered acceptable nominal operations *and* away from all known fault health (diagnostics as defined above) propagation paths.

Prognosis is the assessment of the component's current health and a prediction of the component's future health. There are two variations of the prediction problem. The first prediction type may have just a short horizon time—is the component good to fly the next mission? The second type is to predict how much time we have before a particular fault will occur and, by extension, how much time we have before we should replace it. Or it may be longer term—tell me when to schedule removal of an engine for overhaul. As mentioned above, accurate prognosis is a requirement for implementing CBM.

*Prognostic Reasoner Challenges*

The creation of a prognostic algorithm is a challenging problem. There are several areas that need to be addressed in order to develop a prognostic that achieves a given level of statistical performance.

*What Curve are we on? & Where are we on the Curve?*

The first step in prognosis is determining "where" on the overall health curve the component resides. Along with "where" is "what" fault curve we are on. This is similar to the "fault detection" problem as already discussed above. However the equivalent signal-to-noise ratio (SNR) of the signatures that we are looking for to determine component health will be much lower then for the fully developed fault.

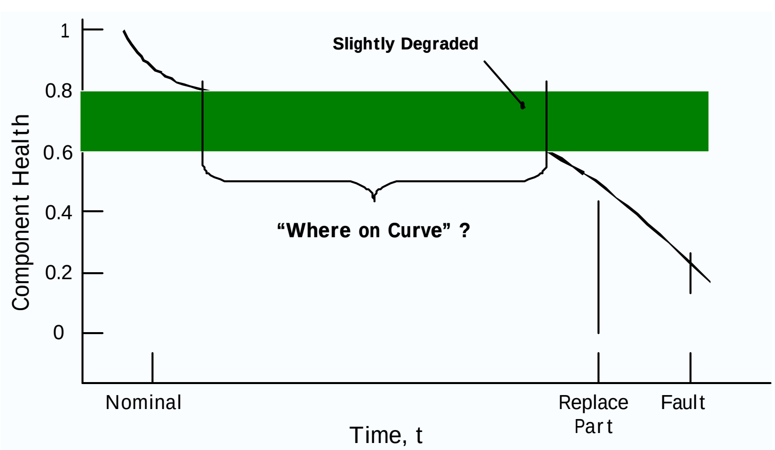



**Figure 4 - What Curve are we on? & Where are we on the Curve?**

This will have two effects. First, because the health component signatures SNR are low, we are always operating in the "gray" area between nominal and a fully developed fault. Because we are in the gray area, even knowing what fault trajectory we are operating on is a challenge. Likely several different fault hypotheses will need to be carried along by the system until a clear-cut condition becomes apparent. Likely a large number of the hypotheses are false so that ultimately no maintenance operation will be required.

Second when we are on the "flat" part of the overall health curve of the component as shown in Figure 4, it is hard to resolve in time where we are on the curve. Again the problem can be attributed because we are operating in the gray region between nominal and a fully developed fault. Suppose that the best we can do in resolving the "health" of a component is to determine that it is in a range of 60-80% of perfect. The component is still quite acceptable. However as indicated in by the green band in Figure 4, we cannot resolve where we are on the curve. Predictions for short time horizons will be reliable (i.e. in determining "good-to-go" for the next mission decisions), but determining remaining life is not possible. The conservative approach would be to assume the worse; that we are at the end of the green part of the curve. Or, we can couple the prognostic with life usage models. The life usage model (assuming one exists) will form the basic estimate of the component health and the prognostic is just used to perturb that basic result.

PROGNOSTICS

Once we determine what the current health of the component is, we need to predict what the health of the component will be sometime in the future. As discussed this prediction can be for a short time horizon or an estimate of the time till the part needs to be replaced or a failure will occur. There are a variety of issues that need be considered. The models that we develop can be of several different forms [Reference: Roemer, et al 2000].

The model will need to accurately predict into the future. Those predictions will be required to be unbiased and to have a small variance in order to be useful. Figure 5 illustrates these problems. In this figure the red line is the prediction of the health of the component from the current state. It does not follow the actual trajectory very well so that it is not that it is a biased estimate of the actual trajectory. However, the model does accurately predict the health / time to replace the component. Is this sufficient?

The green lines represent the error bars for the prediction. The true value of component health

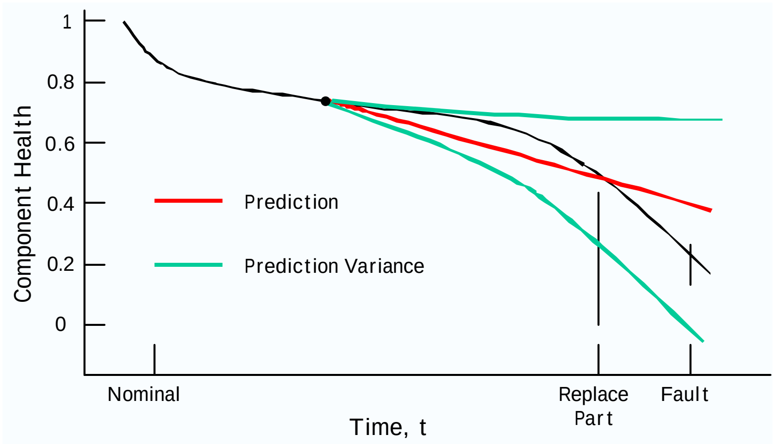



**Figure 5 - Prediction Uncertainty**

curve should fall inside of these error bars as is does. Thus the model is sufficient since it always includes "truth". How useful is it?

The spreading of the error bars defines the time horizon and resolution that can be achieved with this model for performing prognostics. If the error bars spread rapidly then only the predictions are reliable for only a short time horizon. If they are narrow and follow the true trajectory accurately, then the information from the predictions is useful for longer time horizons.

### PROGNOSTIC APPROACHES

There are many different approaches for the development of prognostic algorithms to support the development of prognostic reasoners. For practical purposes, these approaches can be generalized into three basic forms. The first are physical models. These are models that have been developed by experts in the component field and validated on large sets of data to show that they are indeed accurate. The second are systems that embody rules of thumb that have been developed and refined by human engineering and maintenance experts. Examples of these systems are rule-based expert systems and fuzzy logic systems. The third are statistical models that 'learn' from examination of real data that contain nominal and known fault conditions. Examples of these are neural net and data mining systems.

Physical models and rule-based systems contain information for anticipated fault events that have yet to occur on the component that is being monitoring. On the other hand 'learning' systems are good because they can process a wide variety of data types and potentially have performance superior to rule-based system because they exploit the nuances in the data that are not covered by general rules. This is particularly true for new sources of data for which expert analysis, physical models, and rules have not been developed. Physical models and rule-based systems are only as good as the design engineer can anticipate the variety and nature of faults. Learning systems are only as good as the data from which they have been trained. Obviously with the fusion of these systems the best of all worlds can be achieved. [4] [5] [6]

## 3. THE TRI-REASONER IVHM SYSTEM

Next generation Health Management (HM) system architectures must allow for the integration of anomaly, diagnostic, and prognostic (A/D/P) technologies and associated reasoners from the component level all the way up through the aerospace vehicle level. In general, A/D/P technologies are only observers. They observe when a feature is off nominal or damage is accumulating at an accelerating rate. In contrast, reasoners make intelligent decisions about the A/D/P results such as the root cause. Figures 1 and 6 provide a generic illustration of how A/D/P reasoners at each level of a vehicle hierarchy are integrated together. This integration across components, subsystems and systems is vital to isolating the root-cause of failures and propagating up/downstream effects of the faults. While the newest prognostic technologies can sometimes exist without anomaly detection or diagnostics, these are generally essential precursory steps to having a robust prognostic capability in an integrated system. Integration of the individual subsystem health monitoring results can be accomplished with a Reasoner Integration Manager (RIM) that can assess the intra-system A/D/P results to prioritize the most probable fault and recommended maintenance action. A RIM function represents the on-board or ground-based processing module where final decisions about air vehicle health are made. Of course, prudent choices of what processing is performed on-board versus what is transmitted to the ground for post processing must be made a-priori.

The entire aerospace vehicle architecture would actually include several layers of the integrated block diagram shown in Figure 6. For example, each system of the aerospace vehicle (i.e. propulsion, structures, subsystems, etc.) would include its individual subsystems as columns in the matrix architecture, which report up to an Integration Manager at that level. This same architecture can apply for critical subsystem components and many components can exist within a subsystem.

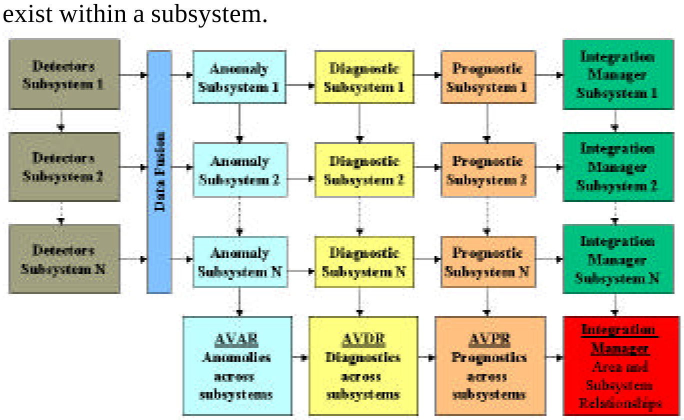



**Figure 6 - The Tri-Reasoner IVHM System**

The functionality and information flow of the HM system architecture can be further represented in Figure 7. When associated with a specific subsystem hierarchy, the information flow begins with the data acquired from the sensor array, which is then validated, potentially fused with other data and key features are extracted from it. The anomaly detection algorithms typically work from the raw data and the associated Anomaly Reasoner (AR) assesses this conditioned information within the integrated model. The concept of the integrated model will be discussed in the next section. The AR's task is to evaluate the raw data and extracted features for correlation and measures of evidence for fault conditions. The correlation and "ripple" effect of anomalies across subsystems is then examined within the Air Vehicle Anomaly Reasoner (AVAR). The AVAR's goal is to correlate anomalies that occur across subsystems and to separate the "upstream" causes from "downstream" effects.

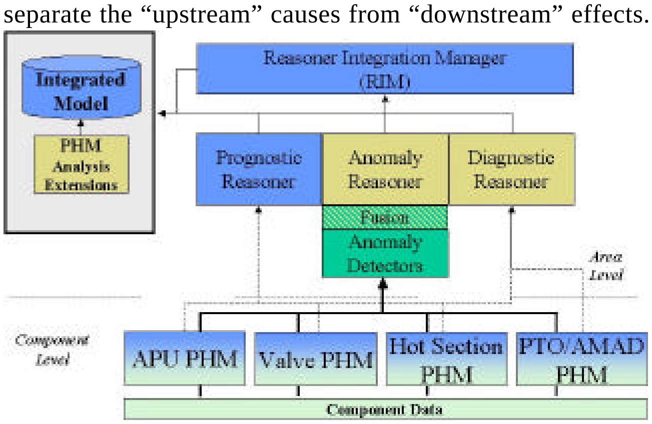



**Figure 7 - Overview of Area Manager Information Flow**

The root cause of an anomaly can be examined further by utilizing the individual diagnostic algorithms and their associated Diagnostics Reasoner (DR). In contrast to the AR, which relies primarily on generic signal processing and statistical techniques, the DR typically will rely on a-priori engineering knowledge and models of a component or subsystem (i.e. model-based diagnostics). Like the AVAR, the AVDR correlates the diagnostic Built In Test (BIT) information and dedicated algorithm results across subsystems.

The individual prognostic algorithms and associated Prognostic Reasoner's (PR) are focused on predicting the time to mechanical failure or conditional failure of a component or components within a subsystem given available HM information [7]. These predictions are given as distributions about a Mean Time To Failure (MTTF), thus resulting in different acceptable risk limits based on the consequences of the particular failure mode. Various levels of prognostic capability exist and this paper primarily discusses and demonstrates model-based prognostic approaches. A PR relies inherently on the individual prognostic algorithm results and an integrated model that is discussed in the next section. Finally, the Reasoner Integration Manager's (RIM) function will keep track of and evaluate the progression of anomalies, diagnoses and prognoses across all subsystems. The RIM will also make the final call on what system users (i.e. pilot, maintainers, engineers) see, do and have access to.

## 4. The Evolvable Aspect of the IVHM System

As the cornerstone of the IVHM architecture, the Integrated Model's accuracy and currency are critical attributes if the conclusions of on-board reasoners are to be relied upon by decision makers. And yet it is impossible to have complete knowledge of all possible failures and their expressions from the beginning of an aircraft's service life. This is especially true since some the aircraft's normal and abnormal behavior will only be exhibited once it is integrated and in use. It is this recognition that has led to the incorporation of embedded learning components within the IVHM architecture. It is likewise impossible that the techniques and technologies used for observing the aircraft's behavior, and for reasoning about these observations, will remain static during an aircraft's operational life. Learning and innovation will inevitably lead to changes in the extent of information and knowledge contained in the IM, the underlying representations, and the algorithms that use them. Since we except that the IM will be matured throughout the aircraft's service life, more rapidly at first and then more gradually, we have considered the nature of appropriate types of technological support for maturation tasks. In this section we present and briefly discuss some of these technologies.

Many design and analysis tools are employed in the engineering of a modern aircraft, even within the specific focus area of maintenance and reliability. Information and knowledge contained within these tools have potential roles to play in the creation and evolution of the IM. Maximizing the usefulness of these resources has led to the creation of a tool integration infrastructure that permits the periodic incorporation of information contained in individual tools into the IM, as well as the interchange of information between tools. The specific capabilities of this infrastructure are thoroughly described in [1][2][3]. Here we will only mention that this technology addresses an important issue that other integration capabilities offer defer to their users--semantic integration. Understanding how information contained in multiple tools corresponds requires either pair-wise associations or else some unified context that provides this interrelation across the range of tools to be integrated. The tool integration infrastructure employed to support the IVHM architecture uses a integrated schema, semantic translators, and tool adapters to move information from tool into this unified representation, and from this unified representation into other tools.

The modeling environment [7] used to support the IM is one such tool. It has many attributes and capabilities that are necessary to support the evolution of the IM. The first is that the representation underlying the IM is described to the modeling environment by means of meta-modeling. The

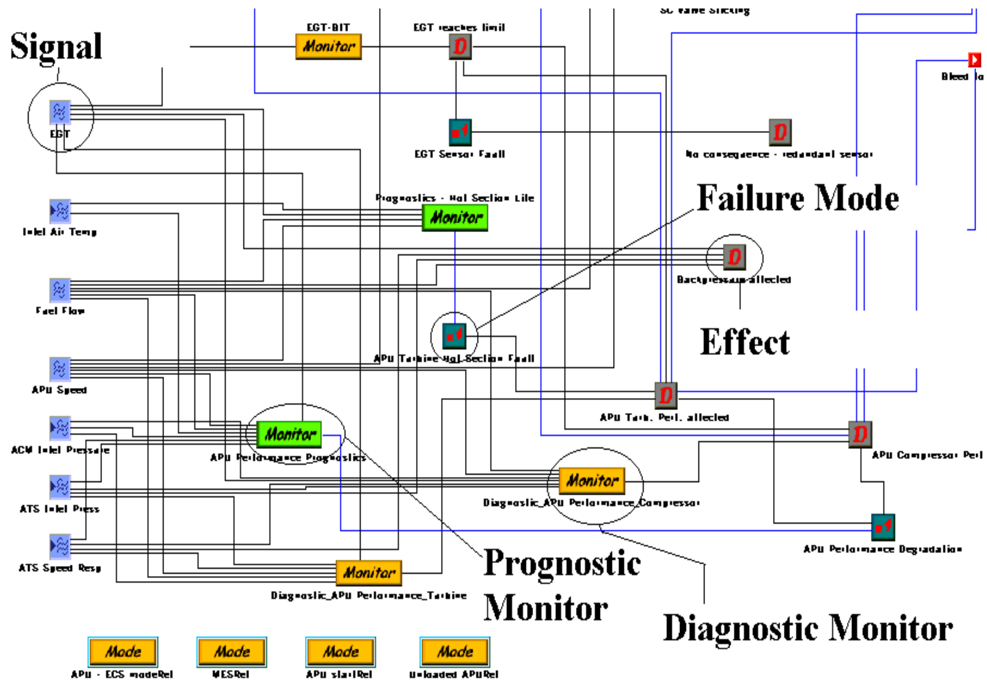



**Figure 8 – This Integrated Model Supplies Information for the Tri-Reasoners.**

environment itself is merely an infrastructure that knows nothing about the particular models being created. This is important because one aspect of the evolution of the IM is that the representation itself will need to evolve in response to improvements in monitoring and reasoning technology, for example. Making the meta-model explicit and keeping it separated from the modeling environment facilitates its evolution without the need to change the supporting infrastructure. [8]

The modeling environment for the IM supports a paradigm known as model-integrated computing. [9] In this paradigm, the integration between embedded information processing components, such as health monitoring and reasoning components of the IVHM architecture, are integrated with their physical environment by means of models. The integration is performed by extensions to the modeling environment, called model interpreters that use the models as instructions to perform the integration and to package information to be used by embedded components. In the IVHM architecture, model interpreters take the design-time representation of the IM and produce the embedded representation, as well as synthesizing the code required to support the application-programming interface (API).

An important issue in creating and evolving the IM is validating the represented knowledge prior to its use on-board the aircraft. The aim of this validation is to ensure that the response of embedded monitoring and reasoning software is predictable with respect to anticipated stimuli. While the ultimate validation of the models may rely on more tradition methods, such as regression testing, we thought that it was essential to provide an interactive form of support for validating modeling decisions that was tightly integrated with the modeling task. We have used the model interpretation interface of the modeling environment to create interactive tools that "wrap" the embedded reasoner implementations. This enables the embedded reasoning algorithms to be invoked directly from the modeling environment to provide immediate feedback for modeling decisions. The observations serving as input to these tools can be recorded, simulated or live. They can be replayed as modelers make revisions to the models until there is confidence that changes create desired responses by the embedded reasoning software. These tools have been produced in correspondence to the kinds of reasoners incorporated into the IVHM architecture. They can be used individually or interconnected, as it would be in their embedded environment. Our early experience has shown these tools to provide invaluable assistance to the modeling task.

## 5. THE INTEGRATED MODEL

The Integrated Model (IM) is an information/knowledge resource that supports all on-board reasoning activities as well as ground-based support functions in the IVHM architecture. It is a graphical associative object representation where nodes represent such things as failure modes, off-nominal conditions (called discrepancies), and the observations made by anomaly, prognostic, and diagnostic monitoring algorithms. Edges represent associations among the nodes, such as failure propagation or incipience. These nodes and edges have attributes that express a priori knowledge, for example the statistical incidence of a failure mode and the likelihood and temporality of failure propagation, or dynamic conditions, such as the status of a health monitor or a discrete diagnostic monitor. Other nodes and edges in the IM can express additional knowledge such as mutual exclusion of failure modes and maintenance procedure references. Model elements can be conditionalized to account for such issues as flight phase or operating modes.

The organization of the Integrated Model parallels that of the physical aircraft, with the hierarchy extending down (at least) to the line-replicable unit (LRU) level.

Interconnections between models, within the hierarchy and among models at a particular level, are expressed in terms of signal paths or physical flows (used to model physics-based interactions, such as a thermal transfer). Failure effects are carried along these interconnections. In effect, the IM is a structural representation, upon which vehicle health information is layered.

The embedded representation of the IM is a subset of the complete IM, containing only that information necessary for embedded reasoners to perform their tasks. Such reasoners access the IM through an application-programming interface (API) that enables the model to be traversed in various ways and provide access to the attributes of nodes and edges along traversal paths. The embedded IM may be distributed among on-board computational resources, with the provision that higher-level reasoners in the IVHM architecture can access enough information to perform their tasks. Since the reasons and means for distribution will vary between implementations, a more complete discussion of distributed access to the IM is beyond the scope of this paper.

The decision to incorporate an integrated model into the tri-reasoner IVHM system has significant implications for the design of embedded reasoners, and it reflects a particular design philosophy for the IVHM System as a whole. Embedded reasoners are presumed not to incorporate into themselves any particular knowledge of the subjects of their reasoning; that is, reasoners are intended to be entirely generic, using only observations and the contents of the IM as the basis for their conclusions. It is possible that certain detection or monitoring algorithms can also be made generic, and the IM can be used to contain the criteria used by such algorithms in making observations. The advantage gained from this approach us clear and vital. If information and knowledge are embedded in the monitoring and reasoning algorithms themselves, then maintenance and evolution become issues dramatically affecting scalability. Keeping track of what information or knowledge is where and how it relates to other knowledge presents and information management problem that, when scaled to up to the needs of modern aircraft, will stress even the most sophisticated information management technology.

This design strategy of model-based reasoning and monitoring algorithms represents a significant difference from the component health management technologies of the past. Realizing it will require a re-examination of techniques and technologies with an eye for issues beyond mere efficacy in a localized context. The IM is intended to represent primarily qualitative and discrete relationships, principally concerning mapping observations about the system to active or incipient failures. However, the attributes provided by nodes and edges in the IM enable the possibility for quantitative information, and it is through these attributes that the continuous behavior of the system and its components can be characterized. It must be recognized, however, that the possibility for generic monitoring and reasoning depends upon the ability to react to quantitative information in a uniform way, or in a way that is itself expressed in the model. Accordingly, the nature of the IM and that of generic monitoring and reasoning algorithms must co-evolve.

## 6. THE REASONER INTEGRATION MANAGER

The integration of the anomaly, diagnostic and prognostic area manager reasoner reports is performed with the Reasoner Integration Manager (RIM). The RIM provides a

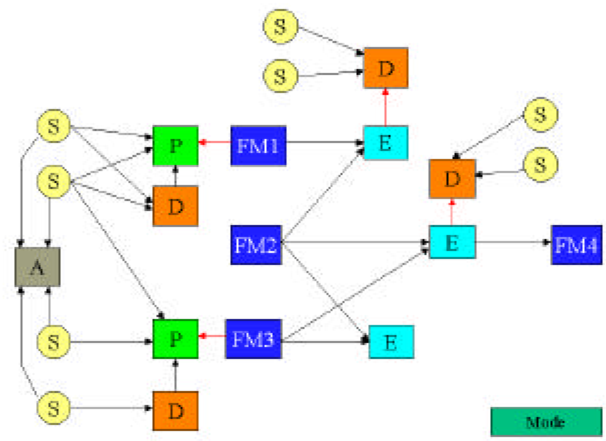



Figure 9 - Generic Representations of Failure Modes, Sensors and HM Technologies

methodical algorithmic process that keeps track of and evaluates the progression of anomalies, diagnoses and prognoses that have occurred across the air vehicle. Through direct algorithm interaction with the Integrated Model and corroborating/conflicting evidence associated with the individual reasoner reports, the RIM prioritizes the most probable fault or failure modes at the air vehicle level. The RIM isolates the most probable failure modes. The RIM creates reports for the operators, maintenance personnel and engineering support staff.

The most significant aspect of the RIM, as is the case for the Anomaly, Diagnostic and Prognostic Reasoners, is the strong relationship it has with the Integrated Model. The connection between the RIM and the Integrated Model stems from how the dedicated A/D/P algorithms developed for detecting and mitigating particular failure modes are linked into the model. This integration can best be described utilizing a portion of an APGS Integrated Model shown in Figure 8. In this figure, the "yellow" Diagnostic Monitors represent the outputs from either dedicated diagnostic algorithms or a result from a diagnostic BIT. These diagnostic monitors are linked to particular failure modes in this Integrated Model based on their ability to either diagnose the failure mode once it has occurred or symptoms prior to it happening.

The "green" Prognostic Monitors represent the outputs from dedicated prognostic algorithms specifically focused on the prediction of a system failure mode. As in the case of the Diagnostic Monitors, the Prognostic Monitors are linked to specific failure modes based on their ability to predict it prior to happening. If a diagnostic or prognostic monitor is triggered during system operation, the associated diagnostic and prognostic reasoners will assess the relevant information within the Integrated Model, which is then used by the RIM to correlate all of the A/D/P reasoner information coming from the system.

A more generic example of how the RIM reasoner interacts with the Integrated Model is given in Figure 9. In this generic Integrated Model representation, the S's represent sensors, A's anomaly detectors, D's diagnostic BIT's or algorithms, P's prognostic algorithms, FM1-FM4 failure modes, and E's effects of the failure modes. In this figure, an anomaly detection algorithm (A) monitors four different sensors (S). If the anomaly algorithm detects an off nominal condition on one of the sensors, because of the Integrated Model connectivity, only Failure Modes FM1 and FM3 are "flagged" as potential failure modes within the Anomaly Reasoner. Failure mode FM2 is not considered a possibility because there is no connectivity within the Integrated Model. Next, if a diagnostic algorithm or BIT were triggered within the health management system that has connections to both FM2 and FM3, the diagnostic reasoner would rank both of these failure modes with equal confidence (with no other information available).

If in addition to these anomaly and diagnostic monitors, a Prognostic Monitor (P) on FM3 revealed a prediction on its mean-time-to-failure (MTTF) that was much shorter than expected, this would allow the prognostic reasoner to highlight FM3 as a concern. The anomaly, diagnostic and prognostic reasoners by themselves would not be capable of seeing the obvious result that failure mode FM3 is called out in each individual reasoners and is therefore ranked highest by the RIM. In the end, the RIM is able to utilize the knowledge from each of these reasoners to make the most informed decision on the systems health. This approach will undoubtedly result in more confident fault isolation and less false alarms.

Finally, let's imagine that Figure 9 represents the portion of an Integrated Model that includes the failure modes associated with a rolling element ball bearing. A physics-based prognostic model of the bearing (P) could be used to calculate the current probability of a failure for a particular failure mode (FM3), and in addition project the future probability of failure based on speed and temperature measurements. However, in this example, let's also imagine that a diagnostic algorithm (D) uses data from a vibration transducer (S) to determine that an unbalance or misalignment condition exists. In addition, this diagnostic monitor also analyzes the vibration features (spike energy or kurtosis) to detect when significant spalling (FM3) of the outer race has occurred.

For the majority of the bearing's life, the diagnostic algorithms do not produce any diagnostic reports and the physics-based prognostic model goes about evaluating remaining useful life based on its usage pattern. However, as the system ages, the diagnostic algorithms begin to detect higher than normal unbalance. With this information, the prognostic model determines that life is being accumulated at a faster than expected rate. The RIM would then be capable of putting together these pieces of evidence to alert the maintainers to examine the bearing at an appropriate time.

## 7. The Integrated Model & Diagnostic Reasoning

The generic diagnostic reasoning algorithm we are initially using is based on the discrete, model-based approach using timed failure propagation graphs (TFPG) described in [10]. This algorithm uses a subset of information contained in the Integrated Model to perform failure isolation by traversing failure propagation paths to associate reported observations of off-nominal conditions (discrepancies) with potential failure causes. The algorithm uses deductive reasoning, based on observation timing, propagation livelihoods, and other evidence to identify a set of candidate hypothesis. Heuristic reasoning is then used to identify and rank the simplest explanations for the reported observations. While this reasoning approach is proven technology, there may be cases where the nature of phenomena, or the available means of attaining the requisite knowledge, within a particular area might suggest alternative approaches. [11] The IM is capable of adaptation to support alternative diagnostic knowledge representations, as well as of forming associations among differing representations.

## 8. The Integrated Model & Prognostic Reasoning

Specific prognostic algorithms are focused on predicting the time to mechanical or conditional failure of a component or system of components given available health monitoring information. These predictions are typically given as distributions about a mean-time-to-failure (MTTF), thus resulting in different acceptable risk limits based on the consequences of the particular failure mode. The job of the prognostic reasoner is to examine the attributes of all prognostic monitors developed across the air vehicle and to prioritize the most probable failure modes to be concerned with. To perform this task, the prognostic reasoner relies inherently on the individual prognostic algorithm results and the Integrated Model previously discussed.

There is a direct relationship between the results generated by the individual prognostic algorithms and the associated attributes of the integrated model (IM). As previously discussed, when the anomaly and diagnostic algorithms detect either anomalous or known fault conditions, their results are examined within the integrated model for relationship to specific failure modes as a part of the reasoning process. In the case of the prognostic reasoner, the individual prognostic algorithms routinely examine the difference between the expected MTTF (expected for "normal" operating behavior) and the actual MTTF (as calculated by the prognostic algorithm). Once a predetermined difference between these MTTF's is reached (based on engineering analysis), the prognostic reasoner "looks" within the Integrated Model to assess how a future failure will effect the overall system operation, as well as determine what it currently means within the RIM.

The prognostic reasoner allows the VHM architecture to continually assess the potential effects of impending failures for critical components as well potential degradation associated with system inefficiencies. Knowing this future risk, the resulting effects and their relationship within the entire air vehicle HM architecture allows the RIM to make informed decisions about future maintenance in a timely manner.

## 9. The Anomaly System – The Key to Maturing the Diagnostic and Prognostic Systems

The anomaly detector is designed to capture, in a buffer, those signals that have been indicted as having deviated from their typical envelope. The anomaly reasoner takes the output from the anomaly detector and correlates this information to the integrated model to trace root cause and functional effects. The reasoner integration manager takes its inputs from the tri-reasoners and answer questions such as: Did the diagnostics system react in a similar fashion to the anomaly and prognostic systems. Whenever the systems agree, there is strong confirmation for the event. If, for example, there is a false alarm in the diagnostic system the anomaly reasoning system would not see anything to corroborate what is being reported to the RIM by the diagnostic reasoning system. This additional information coming from the anomaly system makes these false alarms visible to the maintainer. In similar manner, suppose the anomaly system reacts when the diagnostic reasoning system does not. What does this mean? Is there an impending failure that the diagnostic system will see in due time? Is this an event that only the anomaly reasoning system is equipped to see? If so, is it meaningful to the maintainer? Should a fault monitor be added to the diagnostic system to cover this case? Is there a trend that the prognostic system is not seeing? Should the prognostics system take this into account? Is the control surface motion fundamentally changing? Are the mechanical linkages degrading? Is it a safety threat to the mission, the pilot, crew, and passengers? These questions can be answered in a systematic way, given the evolvable tri-reasoner IVHM system.

The assumptions underlying the tri-reasoner architecture are: 1) the three reasoners have independent outputs, 2) the three reasoners have independent algorithms both at the detection and reasoning levels pursing entirely different goals, 3) the anomaly detection system is highly accurate and robust against false alarms. This last assumption is crucial and a difficult one to meet. We accomplished this through a fusion of outputs from several anomaly detection algorithms. Each algorithm extracts a different feature from the signals it processes. Their fusion produced a robust detection algorithm, as is discussed below.

Fusion technology can combine the results from different processing approaches (such as time-correlation statistics, neural networks, hidden Markov models, and physical models) resulting in superior results. Fusion of multiple approaches has been demonstrated to significantly reduce false alarms while at the same time substantially improving detection and classification performance [13,14,15]. Each group's AD focuses on different aspects of real data signals when performing detection.

Sometimes the detectors are 'complimentary' and support each other's detections. In this case, fusion improves confidence of the detections and thus not only improves detection performance while reducing false alarms. However, sometimes a particular detector focuses on an aspect of the signals not considered by the other detectors. In this case it provides the only anomaly detection. This expands the class of signals that the fused AD is able to process. Fusion has the potential of approaching the goal of perfect detection with zero false alarms.

Table 1 shows a summary of the expected response for the different detectors being developed for advanced military aircraft. The types of anomalies that can be expected are listed on the left. The columns indicate the expected response for each of the detectors; an 'X' indicating that the detector is expected to work well. A '?' indicates the response is not clearly known and depends on the nuances of the data. A goal is to have at least one 'X' in each row. This ensures that no class of anomaly will be missed. However two or more X's ensure increased probability of detection while significantly reducing false alarms.

Table 1. Summary of expected AD detector response

| *Failure Type* | *NNAD* | *BEAM* | *HMM* |
|---|---|---|---|
| Linear transform (gain) | X | ? | X |
| Transient | ? | X | X |
| New 'mode' | X | X | X |
| Feedback | ? | X | ? |
| Sensor failure (in range) | X | X | ? |
| Sensor failure (noise) | ? | X | ? |
| Uncorrelated signals | X | ? | ? |
| Other | ? | ? | ? |

## 10. Appendix: Historical Perspective of Diagnostic Systems for Jet Transports

Jet transports may be in service for over 30 years. There are air-vehicles still flying with health management architectures many technical generations old. The early generation aircraft relied on manual detection and isolation of problems on the ground. These systems were analog and independent of one another. A schematic and a voltmeter were all that was required to troubleshoot.

As these systems became more complicated built in test equipment evolved to warn the pilots of safety critical situations. The maintainer did not use this built in test (BITE). The maintainer still relied on the voltmeter, schematics, and pilot reports.

In time, aircraft design engineers realized that the output of the fault detection monitors could be made available to support mechanic troubleshooting (analog BITE). With these, the concept of "fault balls" was born, and was incorporated on some systems as early as the 1940s. Fault balls are indications, normally on the front of an line replaceable unit, that a fault has been detected - they were originally mechanical, but later were replaced with small light emitting diodes (LED's). In many cases, the line replaceable unit (LRU) front panel contained a test switch to command the LRU to test itself, in a manner similar to how ground support equipment could test the LRU. This capability became known as built-in test equipment (BITE). This capability began to decrease the need for some of the ground support equipment previously used to test airplane equipment. Depending on the system, the fault balls could effectively point the mechanic in the right direction, but schematics and voltmeters were needed for most conditions. However, the BITE of this era was often confusing, not reliable, and difficult to use. Mechanics often distrusted it. Many systems on airplanes such as the Boeing 707, 727, early 737/747, McDonnell Douglas DC-8, DC-9, and DC-10's employed this type of maintenance design.

In the 1970s, some of the increasingly complex systems began to use computers to perform their calculations. This was called digital BITE. With these computers came the ability to display fault detection and isolation information in digital form, normally via numeric codes, on the front panel of the LRU. The digital logic could produce codes that could better isolate the cause of the fault. The digital display offered the capability to display many different codes to identify each type of fault that was detected. These codes often pointed to some description in a manual that could be used to isolate and correct the fault. Many systems on the Boeing 757/767, Airbus A300/310, McDonnell Douglas DC-10, and Lockheed L-1011 employ this approach.

As the number of systems grew, use of separate front panel displays to maintain the systems became less effective, particularly since each LRU often used a different technique to display its fault data. In addition, some of the systems had become increasingly integrated with each other. Digital data buses, such as ARINC 429, began to be used during this time period. Autopilot systems, as they were among the first to use these digital data buses and depend on sensor data provided by many other systems, have been a driving force in definition of more sophisticated maintenance systems. The more sophisticated monitoring was necessary to meet the integrity and certification requirements of its automatic landing function. For example, the 767 Maintenance Control and Display Panel brought together the maintenance functions of many related systems. As the next step, ARINC 604 defined, in 1986, a Central Fault Display System that brings to one display the maintenance indications for potentially all of the systems on the airplane. This approach enabled more consistent access to maintenance data across systems, a larger display than each of the systems could contain individually, and saved the cost of implementing front panel displays on many of the associated system LRUs. In this approach, the CFDS is used to select the system for which maintenance data is desired, and then it routes the maintenance text from that system to the display. This approach was some of the systems on later Boeing 737s, and most systems on the Airbus A320/330/340, and McDonnell Douglas MD11.

Systems continued to become more complex and integrated. A single fault on the airplane could cause fault indications for many systems, even when displayed using the CFDS. The mechanic had little help in determining which indication identified the source fault, and which were merely effects. To solve this and related issues the ARINC 624 was developed in the early 1990's. It defines a more integrated maintenance system that can consolidate the fault indications from multiple systems, and provide additional functionality to support maintenance. Minimal ground support equipment is needed to test airplane systems, as most of this capability is included in the maintenance system. For example, most factory functional tests of airplane systems on the Boeing 747-400 and 777 airplanes consist of little more than execution of selected tests, monitoring fault displays, and monitoring certain bus data using the integrated maintenance system.

The goal in fault isolation on the airplane has always been to identify the single LRU that is the source of the fault. This allows the mechanic to confidently remove the failed component and correct the fault condition. Although in many cases this is possible; there are many others where it is not possible without the addition of sensors or wiring. Addition of these sensors increases the number of components that can fail, and thus sometimes can worsen the maintenance effort. In addition, they add cost and weight to the airplane. There are clearly cases where the addition of such hardware can be beneficial, but the benefits of improved fault isolation must be weighed against the potential reduced reliability, and increased cost and weight of the additional components.

As a result, fault isolation on the airplane cannot practically produce the perfect answer (the single faulty LRU) in all cases. It can point the mechanic to a small group of LRUs in almost all cases. If it is reliable in doing this, it is a very necessary and effective tool to aid in mechanic correction of airplane problems..

References:

[1] Karsai G., Gray J.: Component Generation Technology for Semantic Tool Integration, Proceedings of the IEEE Aerospace 2000

[2] Karsai G.: Design Tool Integration: An Exercise in Semantic Interoperability, Proceedings of the IEEE Engineering of Computer Based Systems, Edinburg, UK, March, 2000

[3] Jeff Gray, George Bloor: Boeing's 3rd Annual Data Exchange Conference, 2000

[4] Brotherton, T.W. and E. Mears, "Applications of Neural Nets to Feature Fusion," *The 26th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 1992.

[5] Brotherton, T.W., T.G. Pollard, and D. Jones, "Applications of time-frequency and time-scale representations to fault detection and classification," *Proceedings of the IEEE-SP Int'l Symposium on Time-Frequency and Time-Scale Analysis,* Victoria, British Columbia, Oct. 1992.

[6] Brotherton, T.W. and R. Mackey, "Anomaly Detector Fusion Processing for Advanced Military Aircraft", *The 2001 IEEE Aerospace Conference,* Big Sky, MT, March 2001.

[7] Ledeczi A., Maroti M., Bakay A., Karsai G., Garrett J., Thomason IV C., Nordstrom G., Sprinkle J., Volgyesi P.: The Generic Modeling Environment, Workshop on Intelligent Signal Processing, submitted, Budapest, Hungary, May 17, 2001.

[8] Karsai G., Nordstrom G., Ledeczi A., Sztipanovits J.: Specifying Graphical Modeling Systems Using Constraint-based Metamodels, IEEE Symposium on Computer Aided Control System Design, Conference CD-Rom, Anchorage, Alaska, September 25, 2000.

[9] Sztipanovits J., Karsai G.: Model-Integrated Computing, IEEE Computer, pp. 110-112, April 1997.

[10] Misra A., Sztipanovits J., Carnes J.: Robust Diagnostics: Structural Redundancy Approach, Knowledge Based Artificial Intelligence Systems in Aerospace and Industry, SPIE's Symposium on Intelligent Systems, Orlando, FL, April, 1994.

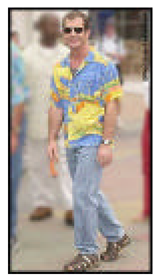
[11] Misra A., Provan G., Karsai G., Bloor G., Scarl E.: A Generic and Symbolic Model-Based Diagnostic Reasoner with Highly Scalable Properties, IEEE International Conference on Systems, Man and Cybernetics, San Diego, CA, October, 1998.

[12] Roemer M., KACPRZYNSKI G., Bloor G., "Development of Diagnostic and Prognostic Technologies for Aerospace Health Management Applications", The 2001 IEEE Aerospace Conference, Big Sky, MT, March 2001.

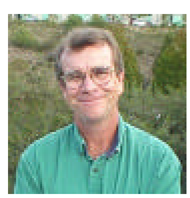
[13] Mackey, R., "Generalized Cross-Signal Anomaly Detection on Aircraft Hydraulic Systems", The 2001 IEEE Aerospace Conference, Big Sky, MT, March 2001.

[14] Brotherton T., Johnson T., "Anomaly Using Neural Networks", The 2001 IEEE Aerospace Conference, Big Sky, MT, March 2001.

[15] Brotherton T., Mackey, R., "Anomaly Detector Fusion Processing", The 2001 IEEE Aerospace Conference, Big Sky, MT, March 2001.

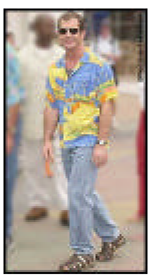


**Les E. Atlas** received the B.S.E.E. degree from the University of Wisconsin and the M.S. and Ph.D. degrees in electrical engineering from Stanford University in 1979 and 1983 respectively. In 1984, he joined the Electrical Engineering Department at the University of Washington, Seattle, where he is Professor and Associate Chair for Research. He co-founded the University of Washington's Interactive Systems Design Laboratory, where he is investigating time-frequency and other signal processing representations for machine learning in acoustical, mechanical, and manufacturing sensor signal processing applications. He also has research interests in auditory processing, speech processing, high-quality music coding, and statistical signal processing. Dr. Atlas received the National Science Foundation's Presidential Young Investigator Award in 1985. He was General Chair of the 1992 IEEE International Symposium on Time-Frequency and Time-Scale Analysis and General Chair of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing. He is a Member-at-Large of the IEEE Signal Processing Society's Board-of-Governors and is Chair of the Society's Technical Committee on Signal Processing Theory and Methods.

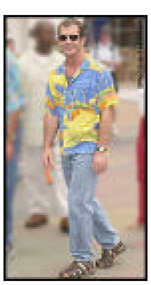


**Tom Brotherton** received his B.S. degree from Cornell University (1974), an M.A.Sc. from the University of Toronto (1976) and the Ph.D. from the University of Hawaii (1982) all in electrical engineering. He was an assistant professor in the Information and Computer Sciences Department at the University of Hawaii in 1983 and with the Orincon Corp from 1983-1999. He is currently a VP of the Intelligent Automation Corp. (IAC) in San Diego; CA. IAC is involved with the development of aircraft and aircraft equipment monitoring software and hardware. Dr. Brotherton is also on the editorial board for the IEEE Press series on Biomedical Engineering. His interests are in the development of adaptive signal and data fusion techniques for machine condition, fault monitoring, and prognostics as well as medical systems applications.

**Ryan Mackey** received his B.A. degree from the University of California at Santa Cruz (1993) for Mathematics and Physics, and went on to an M.S. (1994) and Eng. (1997) in Aeronautics at Caltech. He is presently a senior researcher and charter member of the Ultracomputing Technologies Research Group at the Jet Propulsion Laboratory. His research centers upon revolutionary computing methods and technologies for advanced machine autonomy, specifically deep space missions, UAVs and maintainable aerospace vehicles. His interests also include quantum- and biologically-inspired computing.



**Michael J. Roemer** is the Director of Engineering at Impact Technologies in Rochester, NY and Adjunct Professor of Mechanical Engineering at the Rochester Institute of Technology. He was formerly Vice President of Machinery Analysis at STI Technologies prior to joining Impact Technologies. He has over 13 years experience in the development of real-time, state-of-the-art, automatic monitoring, diagnostic and prognostic systems for large turbo machinery including large steam turbines and gas turbine engines, aircraft, naval propulsion and various other complex systems. He has developed several diagnostic and prognostic capabilities for complex system utilizing probabilistic methods that are directly linked to maintenance planning and system operation. He is the author or co-author of more than 50 technical papers in these subject areas. He is currently the Chairman of the Machinery Failure Prevention Technology (MFPT) Society, a Division of the Vibration Institute, and Prognostics Lead for the SAE-E32 Engine Condition Monitoring Committee.



**Gabor Karsai** is Associate Professor of Electrical and Computer Engineering at Vanderbilt University and Technical Director of the Institute for Software-Integrated Systems at Vanderbilt. He is a naturalized US citizen and does not have security clearance. He got his BSc, MSc and Technical Doctorate degrees from the Technical University of Budapest, Hungary, in 1982, 1984, and 1988, and the PhD degree from Vanderbilt University, in 1988. He conducts research in model-integrated computing (MIC), in the field of development environments, automatic program synthesis and the application of MIC in various government and industrial projects. Since 1988 he has been involved in various projects on fault diagnostics modeling and algorithms, that have been applied and demonstrated in various embedded systems, including chemical manufacturing plants, the Space Station, and next-generation fighter aircraft. His contribution to the project will be in the field of fault diagnostics and prognostics modeling, modeling tools and algorithms, and in model-integration capabilities for the various tools to be used on the project. Currently he is Principal Investigator on various DARPA-sponsored research projects on Fault-Adaptive Control Systems, on Autonomic Logistics, and on Software Synthesis. In the past, he has worked on similar, related projects funded by industry and government agencies. He is a member of the IEEE Computer Society and the TC on Computer-Based Systems. He holds four US patents on run-time infrastructure for model-based systems and real-time fault diagnostics. Some of his relevant publications are:

Padalkar,S, Karsai, G, Biegl, C, Sztipanovits, J: "Real-time Fault Diagnostics Using Hierarchical Fault Propagation Models", IEEE Expert, V.6. pp. 75--85, June 1991.

Abbott, B., Bapty, T., Biegl, C., Karsai, G., Sztipanovits, J.: "Model-Based Approach for Software Synthesis," *IEEE Software*, V.10. pp. 42-52, May, 1993.

Karsai,G.: "A Configurable Visual Programming Environment: A Tool for Domain-Specific Programming", *IEEE Computer*, V.28. pp. 36-44., March 1995.

Karsai, G., Samir Padalkar, Hubertus Franke, Janos Sztipanovits: "A Practical Method For Creating Plant Diagnostics Applications", *Integrated Computer-Aided Engineering*, Vol. 3., No. 4., pp 291-304, 1996.

Misra A., Provan G., Karsai G., Bloor G., Scarl E.: "A Generic and Symbolic Model-Based Diagnostic Reasoner with Highly Scalable Properties", Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, October, 1998.

Karsai G., Bloor G., Doyle J.: "Automating Human Based Negotiation Processes for Autonomic Logistics ", Proceedings of the IEEE Aerospace 2000, CD-ROM Reference 11.0302, Big Sky, MT, March, 2000.

**Mr. Gregory J. Kacprzynski** is a Project Manager at Impact Technologies with 5-yr. experience in the development and testing of diagnostic/prognostic systems for compressors, pumps, transmissions, gas and steam turbines. He has been involved in developing real-time, intelligent health monitoring systems for gas turbine engines for on-wing and test cell applications as well as for other air vehicle subsystems. He has managed multiple SBIR's dealing with next generation condition-based maintenance systems, aircraft prognostics, and health management design for the Navy and USAF. He has published papers and developed technologies in the area of maintenance optimization, FMECA analysis, Life Cycle cost assessment, model-based, prognostics and data fusion technologies. Greg has his MS and BS in Mechanical Engineering from Rochester Institute of Technology.

**Larry Howard** is a Senior Research Scientist at the Institute for Software-Integrated Systems (ISIS) of Vanderbilt University where his research activities principally concern real-time simulation, real-time embedded diagnostics, and learning technologies. Mr. Howard's 20-year career includes work in academia, government, and industry as entrepreneur, technical manager, technologist, and developer.

**George Bloor** is a project manager / lead engineer at Boeing. His current activities include the design of health management systems for ground and airborne applications as well as complex information management systems. George has an MS in mathematical statistics at Iowa State and an MS in Electrical Engineering at the University of Washington.