Integrated Design of On-line Health and Prognostics Management

Mark Walker[1], Ravi Kapadia[1]

[1] *General Atomics Electromagnetics, 16969 Mesamint St., San Diego, CA, 92127, USA*
mark.walker@gat.com
ravi.kapadia@gat.com

ABSTRACT

There are many difficulties associated with the design and implementation of on-line Prognostic Health Management (PHM) systems, including the tardy specification of requirements late in the design cycle, insufficient or conflicting input from pertinent stakeholders, difficult or costly access to domain knowledge, degree of difficulty in validating and testing functionality, and an ill-specified and mostly uncoordinated process. A new methodology is needed that will pull together and coordinate all of the pertinent information obtained from the various system analyses, designers, manufacturers, maintainers, and users at the earliest stages of system design. The methodology should involve a step-by-step process for obtaining information from the various stages of design, and provide mechanisms for modifying design analyses in support of PHM system specification. Additionally, PHM software platforms are needed to facilitate the implementation of these requirements, assist in testing and determining the validity of failure detection and health assessment, and provide powerful model-based reasoning capability that correlates over historical data and across subsystems within an operational context. This paper discusses the issues related to designing and delivering PHM systems, recommends a design methodology that can better address these issues, and describes how the underlying PHM software platform can aid and assist such a methodology to lower the cost, reduce the time to deliver, and increase the quality of next generation PHM systems.[*]

1 INTRODUCTION

Users of mission critical systems expect and require such systems to accomplish their operational objectives with minimal unscheduled interruption (in other words, they would like their systems to be highly available). Meanwhile, the designers, implementers, and integrators of such systems strive to maximize reliability by identifying and minimizing any likelihood of failure. In addition, they are also interested in eliminating or minimizing potentially adverse (and costly) consequences of failure, such as loss of life, equipment damage, reduced efficiency, or harmful environmental impacts.

In support of these objectives, advanced monitoring systems have evolved to better equip the users and maintainers with the data and information necessary to help minimize downtime. Such monitoring systems have increasingly attempted to provide early warning of the onset of failure, with hopes of either minimizing the costs of failure or possibly avoiding them all together. In many cases, they also strive to aid maintainers in optimizing their maintenance strategies according to the concepts of Condition Based Maintenance (CBM) (Butcher, 2000), while also providing automated assistance in troubleshooting and isolation of root causes. These monitoring systems have traditionally gone by various names, including Fault Management (FM) systems, Integrated Systems Health Management (ISHM) systems, Health and Usage Monitoring Systems (HUMS), Condition Assessment Systems

(CAS), Enterprise Health Management Systems (EHM), and Prognostic Health Management Systems (PHM). Regardless of the name, the importance of real-time health and prognostics management for mission critical systems has increased as the users of these systems demand improved operational availability, greater reliability, increased safety, and reduced cost. For the purposes of clarity, we will refer to such systems in the paper as PHM systems.

While the utility of modern PHM systems seems clear, the demonstrated benefits of such systems have been limited, if not disappointing. Developers of PHM systems face many challenges, including difficulties in accessing the data pertinent to defining the proper set of PHM system requirements. In many cases, this data has already been generated through formal design analysis, but is not readily available to the PHM system designers. In other cases, the data is incomplete – or out of context from a PHM system perspective. In still other cases, the data has not been generated – principally because system design processes do not properly consider the specification of PHM system design requirements as being within their scope. As a result, PHM systems have either been prohibitively expensive to specify, or they end up being ill-specified, providing information that is of questionable value to operators and maintainers. Additional problems arise from the fact that PHM system design is often performed late in the design cycle. In the worst cases, PHM systems end up being reduced in scope (or eliminated altogether), resulting in their failure to deliver their promised objectives.

The set of problems associated with specifying the right PHM system for a particular mission critical application is just the tip of the iceberg for the PHM system designer. Once a PHM system has been specified, good tools for delivering the required functionality in a timely manner are seriously lacking. Implementation cycles for traditional software development environments are costly and time-consuming, and provide limited tools for accelerating the time required to deliver tested and validated software that meets the required PHM objectives. As a result, the development of PHM systems has historically involved only limited enhancements to legacy data acquisition and monitoring systems. Usually, these enhancements involve adjustments to alarm thresholds and the inclusion of software built-in-test (BIT) level warnings - all of which intend to provide earlier warning, but result in a familiar flooding of uncorrelated information from disparate sources to operators and maintainers.

Optimally, PHM systems should not only be instrumented for early warning of failures and predictions of component remaining useful life (RUL), but should also provide a complete connection between this information and its impact on overall system health – as well as the user's bottom line. Recognizing the problems associated with developing fully capable PHM software systems, PHM system practitioners have leveraged advanced software technologies (collectively referred to by practitioners as "reasoners") such as signal processing based event detection; artificial intelligence based correlation; expert systems based advisory systems; neural network and statistics based classifiers; neural network and statistics based state estimators; and model-based reasoning (just to name a few) – for the purposes of providing supervisory level intelligence and understanding of the state of the system. Though some progress has been made in the application of these technologies within PHM systems (Schwabacher and Goebel, 2007; Kurtoglu et al, 2008; Patterson-Hine et al. 2005; Schoeller et al. 2007; Kapadia et al. 2009), the problems associated with a lack of available toolsets and the high cost of implementation add to their limited deployment and/or performance.

Additional complications arise for PHM system developers when they attempt to test, validate, and deploy their PHM systems in the real world. Validating system performance using only simulated data has obvious implications, but even with the availability of historical data, many of the techniques being proposed tend to be component or sensor specific, overly sensitive to variances in process, and unable to deal with variances in component wear. These problems tend to result in a limitation of PHM system performance – typically measured in terms of undetected failures (false negatives) and failures which are detected but untrue (false positives). Basically, this means that the PHM system is not only failing to deliver its promised objectives, but is in fact adding to the confusion concerning what is wrong – as well as what should be done about it.

So with all of the obstacles imposed on the designers and developers of PHM systems, it is no wonder that they have had limited success in delivering fully functional PHM systems. It is also no wonder that much of the purse-holding community at large remains reticent to fully embrace the technology.

Fortunately, both the benefits and the demand for advanced Prognostics Health Management capabilities are too well established to be ignored forever. This paper proposes several methodological improvements for PHM system design – along with powerful tools that can help to employ them. The combination can aid in reducing the time and cost for specifying and

implementing PHM system requirements, help enhance the PHM system performance, and help guarantee the delivery of functionality that maps well to what is important to mission critical system users and maintainers. The next section will focus on PHM system design concepts, providing context for section III, which introduces our recommended design methodology. Section IV focuses on the utility of employing model-based reasoning for delivering required PHM system functionality, and Section V provides an overview of the authors' model-based PHM system development environment.

2 PHM SYSTEM DESIGN AND ANALYSIS

Prior to presenting our recommended PHM design methodology, a brief discussion and overview of PHM system design concepts and their difficulty seems appropriate. PHM design tasks to be considered include the derivation of PHM system requirements, the consideration of maintenance and corrective actions, the specification of instrumentation, and the ability for PHM systems to reason over the usage history of entire systems based on operational context. Each of these will be considered briefly.

PHM design methodologies typically include a costly requirements phase that depend on data aggregated from various reliability, failure mode, and criticality analyses on components, subsystems, and systems – tasks which have usually already been performed by various stakeholders at design time, but whose results are spread out between multiple organizations and often buried in formal documentation that fails to address the PHM system objectives directly. Such tasks are the realm of engineering disciplines associated with Reliability Analysis (RA), Failure Modes and Effects Analysis (FMEA), Criticality Analysis (CA), Fault Tree Analysis (FTA), Root Cause Analysis (RCA), Probabilistic Risk Assessment (PRA), and Reliability Centered Maintenance (RCM). These analyses strive to provide the data necessary to reduce risk, guarantee safety, and ensure system performance from both the user and supplier's perspective. Good references on each of these analyses are provided in (Elsayed, 1996; Stamatis, 2003; DoD, 1980; Ammerman, 1998; ONRR, 1981; Stamatelatos, 2002; Moubray, 1997; Okes, 2009).

However, there is a fundamental difference between analyzing system reliability, failure modes and effects, and criticality for purposes of qualifying design and doing the same for the purposes of assessing online operational health. Since the objectives for performing the analyses are different, additional information is necessary in order to determine the appropriate PHM requirements and architecture. The key differences have to do with the consideration of propagated effects and ultimately consequences, as well as the requirements for detecting or predicting them. For system design validation, the principal concern is to identify areas where reliability is questionable, with little or no concern given to specifying the detection of falling capability (decrease in reliability). When deemed necessary, improvements to the reliability of components and subsystems are made through redesign. Redesign might involve replacing components with those of higher reliability or by leveraging redundancy. In other cases, issues of low reliability can be addressed through the use of protective devices.

Where the risk associated with unreliable components and subsystems is determined to be sufficiently low, other strategies are typically considered for improving overall system reliability. These include strategies pertaining to Preventive Maintenance (PM) and Condition Based Maintenance (CBM), but again the recommendations made through design analyses are principally focused on maximizing reliability. Since the primary goal of design analysis is to identify and correct weaknesses and risks in design, little time is spent on specifying appropriate run-time corrective actions or the means by which they should be invoked. PM procedures, initially driven from the analysis of component reliability and associated risk assessment, are intended to help prolong system life at the expense of availability through scheduled downtime. Meanwhile, CBM calls for PHM systems to incorporate on-condition monitoring (Butcher, 2000), thereby avoiding unnecessary scheduled maintenance downtime and directing maintenance activities only for those components which need fixing. On-condition monitoring within PHM systems implies an understanding of the physics of failure as well as the algorithmic approaches for reliable detection of falling capability – neither of which is a direct output from design analyses.

PHM systems should provide timely prediction of effects of detected events so that appropriate mitigating actions can be recommended or even automated. For example, a PHM system might be expected to monitor component deterioration and recommend any appropriate corrective actions. Corrective actions might include system reconfiguration, reduction in output, or the scheduled performance of maintenance actions. In this context, PHM systems – given their potential impact on overall system reliability – are in fact protective devices in their own right. This is rarely

taken into consideration at design time.

Meanwhile, two types of analyses do attempt to bring failure effect propagation and corrective actions into consideration. These are Failure Modes, Effects, and Criticality Analysis (FMECA), an extension to FMEA called out by MIL-STD-1629A (DoD, 1980), and RCM. In the case of FMECA, columns are typically added that also consider maintainability, safety analysis, survivability, logistics support analysis, maintenance plan analysis, and failure detection and isolation (ONRR, 1981). Several of the FMECA considerations are immediately useful for the PHM designer, including failure predictability, failure detection means, and basic maintenance actions. RCM analysis has a similar focus on failure prediction and detection, as well as providing a framework for specifying maintenance actions and plans, but is not strictly based on FMEA. In fact, in spite of its own limited focus on determining and increasing overall system reliability, RCM provides an excellent framework for defining an optimal process for deriving PHM system requirements. The authors present such a methodology in the following section.

Other important aspects of PHM system design include the specification of instrumentation. Typically, monitoring systems are specified based on the minimal set of instrumentation required for detecting specific faults (typically via parameter thresholds). Often, the specification of instrumentation is driven by cost without consideration for the criticality or consequences of specific failures. Furthermore, too often the instrumentation and associated data acquisition systems are specified without a clear understanding of what might be necessary to detect falling capability or to predict RUL. Unfortunately, the cost of adding instrumentation late in the design cycle is often prohibitive. As a result, the PHM system is left with the difficult task of inferring or predicting problems with limited or inappropriate information.

When system failures do occur, the users and maintainers are expected to troubleshoot, diagnose, and repair the system(s) according to the manufacturer's published procedures. This can be difficult and inappropriate when the diagnosis and assessment of health depend on an understanding of the operational history and context. Techniques for considering operational context within event detection and health assessment algorithms typically involve the use of data from modeling and simulation, but the required fidelity of such simulated outputs is often lacking or the simulations fail to take into consideration all interactions from other subsystems. As a result, PHM systems that are tuned according to expected values derived from simulation often perform very differently when presented with actual data. This results in a costly validation cycle that often requires significant modification (or de-scoping) of prognostic behavior.

One way to incorporate operational history and context within a PHM monitoring system is to make use of historical data. PHM systems trained over historical data can provide significant improvements in terms of performance, but are also difficult to develop, train, and validate (not to mention the fact that in many cases, no historical data is available). Only after a sufficient amount of historical data has been collected and analyzed can improvements be proposed to specific instrumentation, algorithms, monitoring systems, and corresponding maintenance procedures. Ideally, PHM systems should be designed to allow for tuning after sufficient historical data can be collected.

Another way to improve the performance and utility of PHM systems is to incorporate integrated electronic solutions that bring together advisements and instruction based on operational context together with technical manuals and troubleshooting guides derived from manufacturer's expertise. Providing insight and automation for operating, troubleshooting, diagnosing, and performing maintenance actions using Interactive Electronic Technical Manuals (IETMs) are just a number of the advanced capabilities being expected from modern PHM systems. It is imperative that such requirements are clearly identified and understood as early in the design process as possible.

It is possible and even likely that changes made during the first few phases of any system design effort will impact the conclusions derived from RCM and FMEA. It is important that the tools used for performing these analyses support iterative design, allowing for modifications and regeneration of results. When the analyses are overly formalized or too narrowly focused, it becomes difficult to identify or understand possible impacts the design changes may have on PHM system specifications.

3 PHM DESIGN METHODOLOGY

The proposed methodology for capturing PHM system requirements is presented in Figures 1-4. The major steps include assessing the results from design analysis; defining the system and its components (assets); performing a functional analysis and associated FMEA; determining the consequences of failure; identifying the associated event propagation and requirements for event detection; considering requirements for usage monitoring; gathering of conditional probability distributions and statistical likelihoods of failure; and specifying all appropriate corrective actions. Many of

these steps are based on classic RCM as described in (Moubray, 1997; NSSC, 2007), but have been tailored specifically for on-line PHM. A brief summary of the major subtasks of this methodology are provided in the following subsections.

3.1 Integrating Design Analysis

The design analyses mentioned in this paper actually cover multiple disciplines and concerns. In a typical critical system design activity, not all of the analyses are being done in an integrated sense. As a result, many of the tasks are redundant or are providing only pieces of the information needed for specifying PHM requirements. Meanwhile, the steps involved in PHM design require very similar analyses to be performed. If the data required by PHM designers is not available from the various reports and documents derived from system design, it is likely that the PHM designers will need to perform the analyses again – usually a prohibitively expensive exercise. Even if the data is available, experience shows that the largest expense in time and money associated with determining PHM requirements is tied up in knowledge capture. Knowing what data is needed and where (or from whom) it can be found can be difficult and time consuming. This suggests a PHM system design methodology that aggregates the data resulting from multiple design analyses and integrates it with the specification of PHM system requirements. Since the data from these analyses is required for specifying PHM behavior, integrating the design, safety, operational diagnosis, and maintainability analyses so that the activities only have to occur once presents an opportunity for considerable cost savings.

The activities associated with the aggregation and integration of design analyses with PHM system specification involve significant culture changes. One of these changes entails a tailoring of the analyses associated with RA and RCM (including the data derived from PRA, FMEA, and CA) so as to reduce redundancy and ensure that all necessary data is collected and provided to the PHM system designers at the earliest opportunity and in an appropriate format. This is not a trivial undertaking, due to the large number of organizations involved and the degree of formality associated with their processes. Attempts are being made by the authors to define such an optimal process, but further work needs to be done.

Another required culture change relevant to improving PHM design is the coordination of design analyses with the design of PHM. Clearly, it is preferred to specify PHM system requirements as early in the critical system design process as possible. This seems a novel idea, but there have been a few attempts at specifying such a unified process (Brignolo et al., 2001; Kurtoglu et al., 2008). The benefits go both ways, as the formal process involved with specifying PHM requirements can be of great value to the system designers in analyzing their design and assisting in the specification of instrumentation. Still, the principal roadblocks to a common process include a silo-based failure to communicate across disciplines, and a difficulty in securing good PHM requirements early while the design is still in flux.

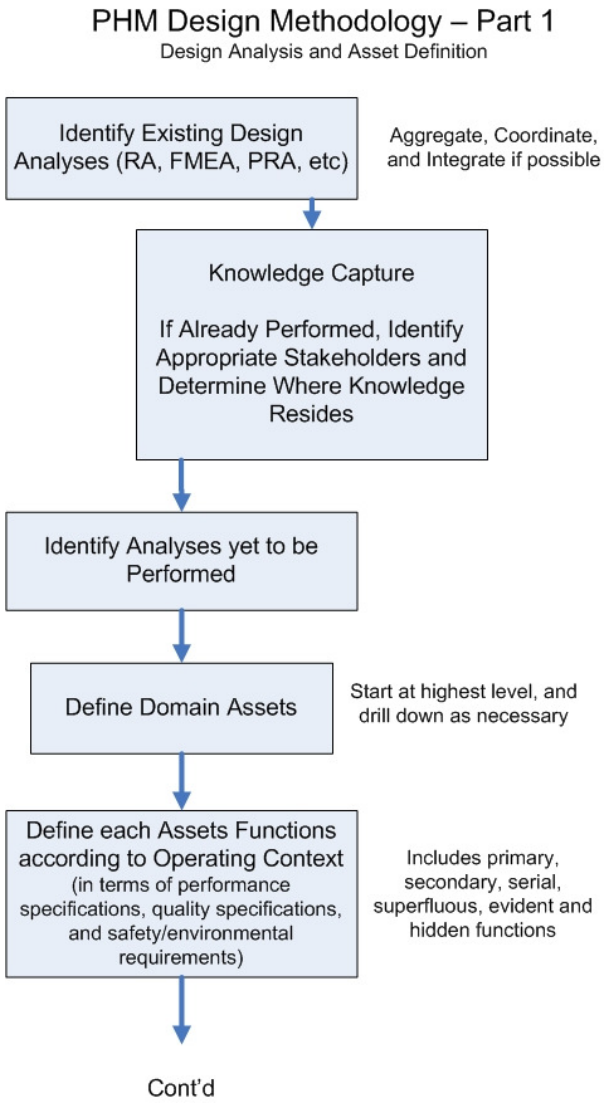

Figure 1: PHM Design Methodology – Part 1

3.2 Defining the Assets

The second subtask in our PHM design methodology is to properly define the system assets (components) being monitored (refer to Fig. 1). This task involves the

review of all available and appropriate system documentation, even though the design may be premature in nature. These include architecture diagrams, functional block diagrams, schematics, interface descriptions, system specifications, and other related design documents. It is imperative that a consistent representation of the system and its architecture is formulated. This representation serves multiple purposes and will: define the assets covered by the PHM system (which are also the assets for which FMEA should be performed); define the underlying software object model over which the PHM system will reason; provide a means for communicating what is being modeled back to the designers; provide a means of determining to what detail the system should be modeled; and provide a means by which the subsystem interactions will be defined and understood.

3.3 Defining Functions

The third step in the proposed PHM design methodology involves defining the various functions of each of the assets defined in step 1. According to RCM (Moubray, 1997), it is important to specify asset functions according to their operating context – that is, according to their detailed performance specifications. This may seem obvious, but most FMEA exercises involve the analysis of equipment failure modes that are independent of how the asset is expected to be used. From a design perspective, basing functional descriptions on operating context should help alert the designer whenever desired performance exceeds or is marginally close to the initial capability of the asset. It also helps to ensure that all stakeholders in the design process share a common understanding of what the asset functions are.

As will be seen in the following section, basing functional descriptions on operating context also guarantees that detected failures and propagated effects of the PHM system will be specified according to what the user cares most about. This is the essence of the RCM based methodology for specifying PHM requirements: to deliver what is required, and avoid superfluous fault modeling and event detection.

As a result, it is increasingly important for PHM systems dedicated to monitoring how well a system is performing (as well as predicting how it is expected to perform) to take all appropriate performance metrics into account. Requirements for PHM systems should therefore be based on the reliable and early detection of any failure mode that is expected to interfere with the specified functions of the monitored system(s).

3.4 Defining Functional Failures

The fourth step in the PHM design methodology involves the specification of each functional failure – or ways the described asset's *functions* might fail (refer to Figure 2). Like the functions, the functional failures should be written within the operational context. These include specifications for performance, quality, safety, and environmental impacts. Again, one of the benefits of defining failures according to function is that all stakeholders can share and agree on what constitutes a functional failure – as well as its criticality. It will then be the list of functional failures that defines what will be analyzed during the Failure Modes and Effects Analysis.

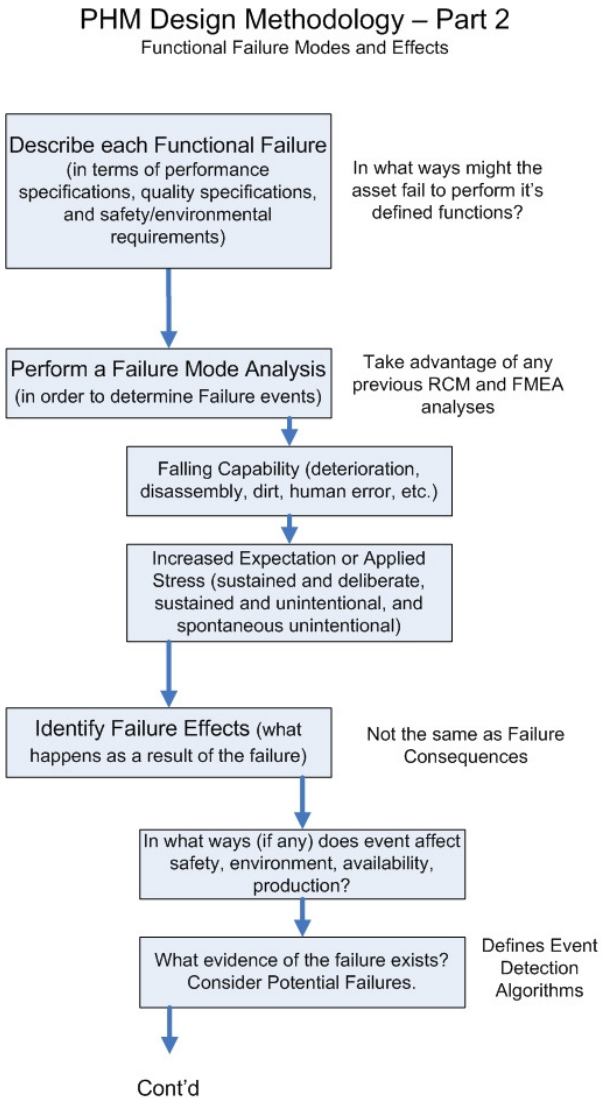

Figure 2: PHM Design Methodology – Part 2

One of the interesting aspects of focusing on functional failures is that sometimes there are more

functional failures than discrete failure modes of components. This highlights the need for consideration for separate ways that the system and its components may fail to perform their desired objectives – as well as considering each individual consequence. Sometimes, component failures result in no failures of function. An RCM focused methodology helps to identify such cases where consequences are minimal, and where minimal monitoring may be acceptable.

3.5 Performing FMEA

The fifth step in the PHM design methodology involves the very important task of performing an FMEA. Since reliability analysis and FMEAs are likely to be required as part of system design, it is critical to try and coordinate with those responsible in order to avoid redundant effort. The specific steps for performing an FMEA are provided by many sources, including (Stamatis, 2003) and (DoD, 1980), and are not reproduced here. In any case, it is imperative that the FMEA be performed very early in the design cycle.

Among other things, FMEAs help to identify failure symptoms (effects) – potentially measureable evidence that a failure is either occurring or is on its way to occurring. Resulting detection algorithms are responsible for generating failure *events* – asynchronous conclusions drawn by the PHM system that will invoke additional PHM behavior. These events include those events typically detected by thresholds placed on measurement parameters, but also include more complicated methods of measurement and detection for falling capability (caused by deterioration, disassembly, foreign materials, or even human error), increased expectation (cases where the required or expected performance levels might change), and applied stress (cases where the asset is being subjected to stresses outside of expected or sustainable values). Identifying potentially measurable evidence also helps to specify the required instrumentation – as well as mechanisms for detecting failure effects. Meanwhile, advanced PHM systems require advanced tools to assist in health monitoring and event detection. Model-based reasoners have demonstrated some key advanced PHM capabilities relative to monitoring usage, detecting events, propagating events, correlating events, and isolating root causes. Some of these capabilities will be discussed in the following sections.

3.6 Consider Consequences

The next step in the PHM design methodology is identifying the consequences of system and component failure (figure 3).

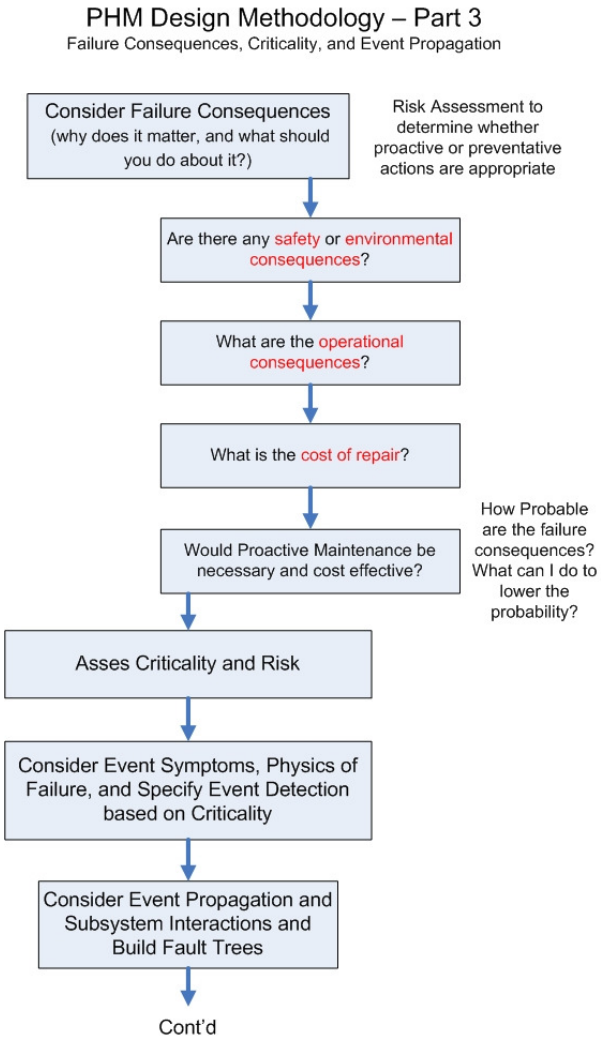

Figure 3: PHM Design Methodology – Part 3

Consequences go beyond the measurable evidence of failure, and consider in what ways the effects will matter to the users. The consequences of system and component failures are often completely overlooked by traditional FMEA and Reliability Analysis, even though these consequences are directly related to the user's bottom line. Specific consequences within the operational context such as failing to deliver to customer expectations, lost productivity, permanent equipment damage, safety hazards, and environmental hazards are ultimately the things that the users care most about. Criticality Analysis and Probabilistic Risk Assessment both attempt to address these concerns by focusing on potential consequences and their likelihood of happening. Critical failures are those failures deemed to have dire consequences – mainly from operational, safety, and environmental perspectives. Meanwhile, PRA attempts to identify likely risks, and

mitigate them through various means. PHM designers should therefore make use of the valuable insights provided by these types of analyses.

3.7 Defining Event Propagation

There are a number of other tools that have been used to assist in the generation and understanding of FMEA data. These include Fault Tree Analysis, which organizes the data in terms of a branched tree, and provides more information on the cause and effect relationships between various failure symptoms.

The key for the PHM system designer is to have a good understanding of how failure events propagate through the system early in the design cycle, thereby allowing the PHM system to incorporate appropriate correlation logic. When using model-based reasoners, the authors have demonstrated the utility of employing software modeling tools that can capture fault tree logic graphically, which provides for powerful propagation and graph traversal capability at run time that supports fault isolation and root cause analysis (Kapadia et al., 2009; Walker, 2007; Walker et al., 2007). The same tools can assist in propagating predictions of falling capability, when sufficient evidence exists. In any case, the ability to graphically model the failure modes, effects, and consequences at design time has proven to be very useful. Stakeholders can examine and validate the underlying fault models early and with ease. Any necessary changes can be made very quickly by editing the fault model graphically, and the propagation behavior can quickly be tested and validated using the same toolset.

3.8 Gather All Distributions

While many of the failure events defined for a system involve simple, threshold-based detection means, very often the successful prediction of health problems requires the identification and understanding of the statistical likelihoods of failure for each of the assets being considered (see Figure 4). These likelihoods are typically specified by conditional probability distributions, and predictions involving the onset of failure require manipulations and calculations involving these probability distributions. It is important to identify tools which support this type of reasoning.

The first step in managing this aspect of PHM is to identify and gather all of the necessary statistical information. This can be time consuming and costly, depending on where the information is located. In many cases, good information about the likelihood of failure for various failure modes is not available at all. It is imperative to know which of these pieces of information is actually required, based on the steps outlined above (e.g. which failures are critical, and what are the potential consequences?), so that time is spent only gathering the data that is pertinent to the users.

3.9 Monitoring Age and Usage

Clearly, time varying processes are an important aspect of PHM, but they are often ignored by the design analyses used to derive specific health monitoring requirements. Typically, failure modes and estimations of component reliability are calculated as static values or distributions. In fact, the reliability of most systems and equipment are a function of age, usage, and operating mode - and therefore time. For example, the statistical distributions associated with time to failure, time to detect, time to abort, and time to repair, are constantly changing. Not only are they a function of the equipment life cycle, but they are also impacted by the stresses applied by external processes and by subsystems which share common interfaces.

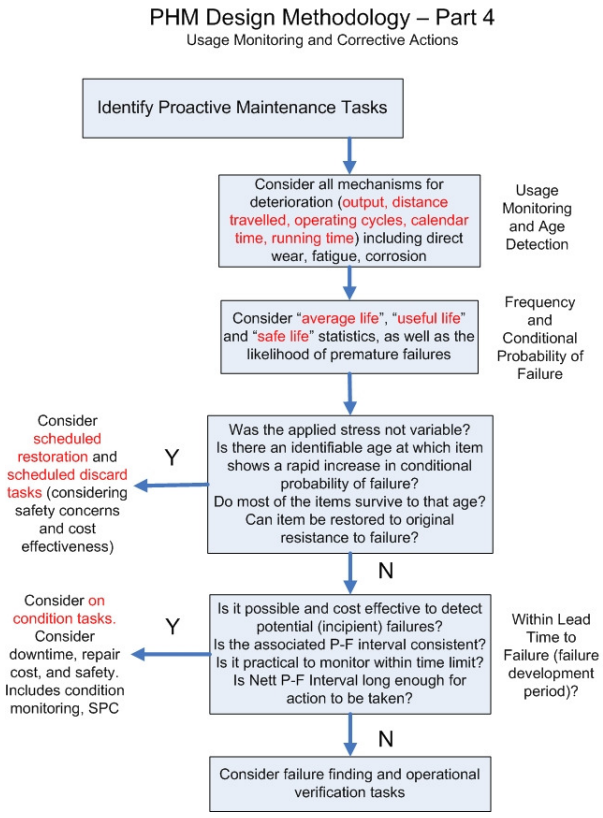

Figure 4: PHM Design Methodology – Part 4

Since PHM is responsible for monitoring expected equipment life, it is important that the dynamic behavior and utilization of the components and systems are also managed and monitored. As a result, it is

imperative that the dynamic nature of the data be taken into account at the time the design analyses are being performed. Furthermore, the PHM software platform (particularly any reasoners) should be able to account for and reason over the dynamic nature of the state of the system.

3.10 Gather Other Design Criteria

In addition to the probability distributions for the various functional failures, the PHM designer will need to identify other pertinent design criteria for the various assets being considered. In order to monitor usage and perform stress detection, the PHM system will probably need to know what the operational criteria for a particular asset might be. For example, the performance specifications for a vehicle engine might only make reference to sustained vehicle speeds, and not specify the preferred operating regime in terms of engine rotations per minute (rpm). What if the engine is operated at high RPM briefly, but repetitively? What is worse - long periods of stop and go, or continuous operation at high RPM? Basically, the PHM designer should be asking questions like: "Under what conditions should the estimated likelihoods of failure be updated at run time?" These questions need to be asked early in the design cycle, so that sufficient time can be granted for obtaining the necessary information.

3.11 Identify Proactive Tasks

In the spirit of RCM, the final step in the PHM design methodology calls for identifying any proactive task that can help in mitigating risk or increasing overall system reliability. The principal reason for this step is to provide a better understanding of what can be done to help mitigate failures and increase overall reliability. It is also designed to help identify the maintenance actions that are possible, necessary, or desirable. In response, the PHM system will need to be instrumented to identify at run time the need for such proactive maintenance tasks.

Before proactive tasks are identified, all mechanisms for deterioration are considered – and the means for performing age detection on the various assets is identified. Required instrumentation and algorithm performance is also noted. The conditional probabilities of failure that were collected in the previous steps are taken into consideration – particularly with respect to measuring the remaining life of components.

This step is also where stress detection is formulated and proposed. After stress has been assessed (which is usually a cumulative effect), the likelihoods of failure can be modified accordingly, and predictions presented with respect to functional failures and their consequences. The PHM designer then asks the question: "What maintenance activities might be scheduled (and when) that might restore the asset to its original resistance to failure?"

In addition to making predictions based on changes to an asset's expected failure, the PHM designer then focuses on the physics of failure and considers any possible algorithms that can be employed to detect the onset of failure. Cost analysis for the detection means is performed at this step, and the utility of on-condition monitoring is considered.

If none of these techniques are possible (in other words, predicting failures is mostly impossible and/or failure rates are mostly random), then the PHM designer must consider appropriate inspection tasks that might be performed in order to help mitigate any consequences of failure. These types of maintenance tasks are usually expensive, but if the cost of failure is unacceptably high, then such tasks are appropriate.

4 MODEL-BASED REASONING

Concern about mission critical system availability, reliability, and performance has spawned the development of powerful reasoning systems that attempt to assess overall system health, detect events that suggest degradations in health, make predictions about the expected availability and RUL of the system and its functions, and provide advisement regarding the steps that can be taken to avoid any negative consequences of failure. As mentioned earlier, these systems have evolved from basic diagnostic systems to advanced Prognostics and Health Management systems.

Some of the key enabling technologies being employed by PHM reasoners include expert system based decision support, advanced digital signal processing, Bayesian Belief Network based data fusion, neural network based function approximation, Kalman Filtering and state estimation, clustering and statistical based pattern recognition, rule-based inferencing, and model-based reasoning. Many of these technologies are being investigated by PHM practitioners, and several are the focused domain of specific tool vendors (Schwabacher et al., 2007; Kurtoglu et al., 2008; Patterson-Hine et al., 2005; Schoeller et al., 2007; Kapadia, 2003; Kapadia et al., 2007; Kapadia et al., 2009; Walker, 2007; Walker et al., 2007). The maturity of several of these approaches is increasing, while others require additional research and validation. While many of these reasoning technologies are being applied successfully for detecting degradations in

component health, few are being employed for correlating events across subsystems or providing insight into overall system level health.

Our focus here is on the utility of model-based reasoners relative to their ability to provide overall understanding of system state and to aid in employing policies based on mission objectives and operational context. The ability to model across the processes of an entire enterprise provides higher-level reasoning mechanisms that take advantage of reasoning over operational context and aid in improving operational efficiency; providing situational awareness; enforcing policy and procedure; and delivering the life-cycle objectives of CBM and RCM.

There are several key benefits to applying model-based reasoning within PHM system architectures relative to lowering the total cost of implementation. One benefit focuses on software reuse of functionality that is common and generic across applications. For example, much of the evaluation of operational health and prognostics for system components depends on logic that is independent of the system for which they are being utilized. Much of the component-level analysis for these pieces of equipment can be defined generically and readily applied to new applications with little to no modification. This feature also deems the model-based framework exceedingly scalable in that as new components and equipment are added to the system, a minimal amount of new code is required to include them in the overall management strategy.

Another benefit of applying model-based reasoning to PHM system design comes from its ability to apply expert reasoning over a centralized object model that takes into account operational context and whose model properties are both monitored and simulated generically according to first principals. In this way, the system can generate state predictions, compare these predictions to real-time measurements, and then exercise policy driven procedures and processes that are designed to respond to sub-optimal system states according to expert knowledge. Another related advantage of model-based reasoning comes from the ease with which applicable human expertise can be incorporated into the behavior of the underlying generic object model.

A third benefit of applying model-based reasoning within PHM is the ease with which new functionality can be added to existing equipment classes (and the ease in defining new ways that these classes might relate to one another). This "extensibility" encourages iterative deployment of functionality based on cost and need, while allowing the implementation to focus principally on what is being delivered. Once the underlying object model has been created, adding new features and functionality that apply to that model and its classes are greatly facilitated by the use of model-based reasoning.

4.1 Domain Modeling

For system-wide PHM applications, we have found it essential to create a system level object model in software over which the PHM application will reason. This model is built from the data derived in step 2 of the PHM design process. Special care must be taken to ensure that the underlying object model is consistent with all other system specifications. This ensuring of consistency can be greatly facilitated by step 1 of the PHM design process.

Typically, a hierarchical domain model is produced that will consist of higher level system and subsystem objects, along with the specific instances of equipment that are contained in those systems. All of the domain model objects are defined according to a reusable generic class hierarchy.

The object model is further enriched through the specification and population of object attributes, many of whose values will be linked to incoming real-time sensor information and configuration data. Other attributes include operating parameters such as run-time, cycle time, operating mode, etc. Also included in this model are the definitions of relationships that might exist between objects, as well as the specific (and often dynamic) instantiation of these relationships.

4.2 PHM Layered Architecture

In order for PHM systems equipped with model-based reasoners to deliver the full functionality required by the PHM design methodology, a layered software architecture is useful. The authors have proposed such a layered architecture, depicted in Figure 5. Included in this architecture are five layers, including the input layer, the configuration data layer, the event detection layer, the fault management layer, the health management layer, and the prognostics layer. The layering also conforms to Open Systems Architecture for Condition Based Maintenance (OSA-CBM) (Puri, et al., 2006), providing for communications and API's within each layer. This allows the PHM system to be implemented with various COTS tools and applications.

The Input Layer includes the measurement and state information required to properly assess system health. This layer includes all validated sensor measurements, along with state information pertaining to operating modes and commands. This would include valve and switch state changes that prompt an update to the

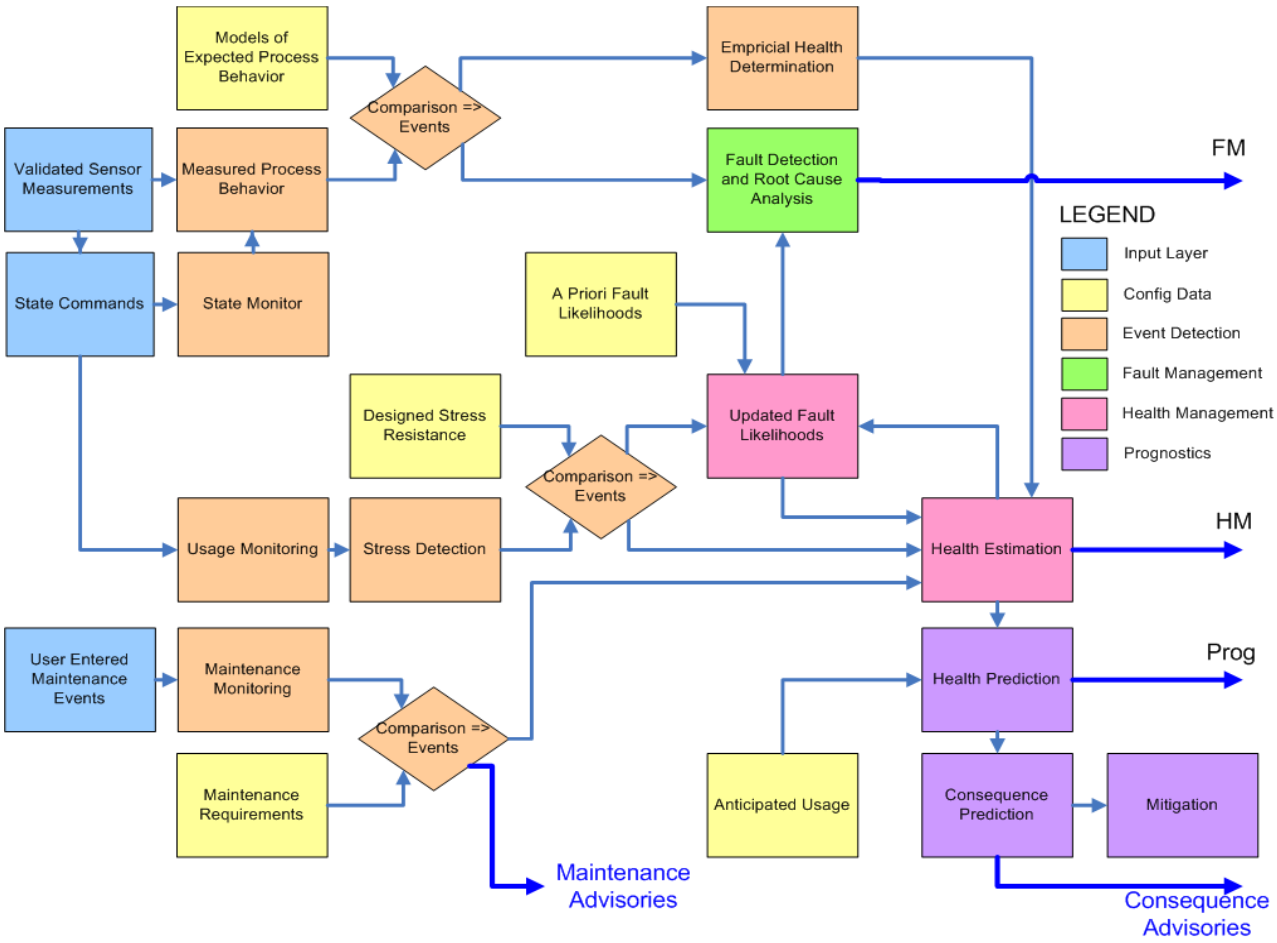

Figure 5 PHM Reasoner Layered Architecture

underlying domain model within the model-based reasoner. The configuration data layer provides both a placeholder for, and the ability to make use of the system specific configuration data necessary for providing operational context. In addition to detailed component specifications (weight, size, volumes, temperature specifications, etc.), configuration data might also include models of expected system behavior, a priori fault likelihoods, designed-to stresses, maintenance requirements, and anticipated usage.

The Event Detection layer is responsible for making comparisons between measured and expected process behavior. It is also responsible for monitoring state information and transitioning the object model according to state commands and operational modes. Similar comparisons are performed by the event detection layer relative to stress detection and maintenance monitoring. For stress detection, it is necessary for the system to perform usage monitoring, and make comparisons between detected stresses and the designed-to stresses. For maintenance monitoring, comparisons are made between monitored maintenance activity and maintenance requirements. The Event Detection layer is also the place where data driven health assessment is performed – as with neural network or clustering based anomaly detection.

The Fault Management layer focuses on isolating faults and making root cause determinations, while the Health Management layer makes a determination of health based on all available information. The Prognostics layer takes the estimates of health and makes predictions and recommendations accordingly based on anticipated usage and criticality of consequences.

4.3 FMEA and Fault Models

According to any FMEA performed, the discrete number of generic failure modes can be identified for the system, subsystems, and components within a PHM domain model. From these failure modes it is possible to construct a generic operational fault model – a directed graph which depicts the cause-and-effect relationships between the component's failure modes,

the upstream root causes, and any of the observable (or measurable) downstream effects. For this, we use a graphical programming language that can both represent the information provided by FMEAs; as well as provide context for the software to perform propagations at run-time. Since the propagation of fault events and predictions at run-time are operational in context, some care must be taken to ensure that the FMEAs are defined with PHM objectives in mind. We have found FMEAs that are focused only on verifying reliability of design to be flat and lacking the detail necessary to capture the event propagation that occurs during operation. In the worst cases, the failure modes and the effects are identical (failure mode: function A fails; effects: function A fails). In these cases, some modification is required for translating design FMEAs and FTAs into operational fault models.

Within a PHM system, such generic fault models can be traversed by software to aid in determining the causes of abnormal system behavior. The models can also be traversed for predicting the downstream impacts. By traversing all applicable fault models upon receipt of detected events, PHM software can identify the necessary tests to diagnose and isolate the root causes of problems, ruling out other possible explanations that are not substantiated by event data. This is known in industry as *Root Cause Analysis* (RCA).

Similarly, by traversing downstream in a fault model, PHM systems can predict degradations and the onset of failure – commonly referred to as *Impact Analysis*. Together, these analyses provide a powerful diagnosis and prediction capability that fully leverages the generic nature of the object and relation definitions within a model-based PHM application (Kapadia et al., 2007; Kapadia et al., 2009; Walker, 2007; Walker et al., 2007).

4.4 System-Wide Event Detection

It is important to ensure that each observable event is detectable and distinguishable through a properly specified method of event detection. These events typically represent early warning of deterioration or falling capability (Moubray, 1997).

Historically, PHM systems have focused primarily on single component health monitoring based on sensor measurement thresholding for detecting specific equipment failure modes. These techniques are expected to generate alerts and alarm notifications whenever system parameters begin to deviate from normal (as determined from statistical analysis of historical data), deviate from expected values (as determined from modeling and simulation), or trend towards some a priori classified failure signature.

Both FMECA and RCM analysis provide a lot of the information needed for PHM event prediction and detection. According to MIL-STD-1629A, failure predictability (which is annotated as part of the FMECA-maintainability charts) should include "information on known incipient failure indicators (e.g., operational performance variations) which are peculiar to the item failure trends" (DoD, 1980). This information, which specifies what data is necessary and how it should be processed to predict the failure, aids in implementing PHM system failure prediction algorithms.

Another category of information provided by FMECA and RCM is the failure detection means. This information explains how each failure mode can be detected, provides methods for resolving ambiguities (cases where more than one root cause exists per failure mode), and provides detailed information on appropriate monitoring or warning devices. Obviously, this information is directly applicable to the specification of PHM system requirements.

Often missing from PHM system event detection is the ability to correlate sensor data across multiple subsystems according to operational context. One of the reasons for this limitation is the component specific focus of hardware FMEAs. Though functional FMEAs are limiting in their ability to provide specific information about the failure modes of equipment instances, they typically provide better information about how failure events propagate across subsystem boundaries. The RCM methodology for specifying PHM requirements takes this into account, while an object-oriented domain model enables the PHM system to reason about such subsystem interactions.

4.5 Health and Prognostics

The assessment of health and making of predictions regarding RUL are inextricably linked to system specifications and dependant on the outputs of the PHM design methodology. As a result, the health management and prognostics layers need to be exceedingly flexible. For health management, it is important that any PHM system have the ability to make use of all information relative to the state of the system. Such information includes the probability distributions for each failure mode, along with knowledge regarding how those probabilities might change dynamically when presented with excessive stresses. Stress detection implies that appropriate instrumentation is in place – a requirement that should

be supported by consequence and criticality analysis.

Health assessment and prediction may also be a function of monitored maintenance activity (or lack thereof). In one case, the likelihood of failure may increase immediately following maintenance activities, while in another case, delinquent maintenance activity might suggest a higher likelihood of failure. In any case, health assessment is usually accompanied by confidence indices. The degree of confidence in health assessment can be modified by additional evidence – a fusion task well suited for Bayesian Belief Networks. When PHM reasoners are equipped with empirical health assessment algorithms, such as neural networks, these estimates can be used by the health and prognostics management layers to increase their confidence in their assessments.

4.6 Maintenance Actions

One of the purposes of both the FMECA and RCM is to identify any appropriate maintenance actions that should be taken in the event of a predicted or detected failure. Maintenance actions include those that are preventative, those that are corrective, and those that involve some form of inspection. Though maintenance actions are typically manual operations, there are many instances where these actions can be automated. Our PHM methodology provides insight regarding automated maintenance actions, as well as details about procedures, costs, and distributions of time-to-repair. A model-based reasoning platform coupled with a supervisory system that automates decision support can assist in making these maintenance decisions. Such a software platform can also be used to integrate with electronic technical manuals and other forms of electronic media useful to the maintainers.

5 EXAMPLE PHM ARCHITECTURE

Based on several years of work implementing PHM systems for mission critical applications (Kapadia, 2003; Kapadia et al., 2007; Kapadia et al., 2009; Walker, 2007; Walker et al., 2007), the authors have implemented a framework of software components that are designed to facilitate the development of PHM systems according to the methodology presented above. This framework is collectively referred to as the Health Monitoring, Assessment, and Prognostics engine (HealthMAP™)

The architectural approach of a typical HealthMAP™ based solution is depicted in Figure 6. At the core of the architecture is a reasoning execution engine, which is itself a real-time virtual machine capable of scheduling, simulating, inferencing, trending, and multi-threaded processing. The reasoning engine is an OSA application that supports standards-based interfacing, allowing it to communicate directly to various applications and real-time data sources. These might include sensors, transducers, programmable logic controllers (PLCs), distributed control systems (DCS), supervisory control and data acquisition systems (SCADA), data aggregation platforms, and even third-party management applications. The engine also supports interfacing to advanced dynamic modeling software executables. This is especially useful when the application requires state prediction that involves higher order mathematics. Additionally, the engine typically interfaces to any number of database systems, such as personnel, configuration, maintenance, and inventory databases, as well as historical data. An Open DataBase Connectivity (ODBC) compliant database is also typically incorporated for the archiving of the events, maintaining persistence, and managing diagnostic conclusions.

Other possible sources of input data include business process and policy information: including data from technical manuals; troubleshooting procedures; maintenance procedures; and casualty response procedures. Each of these can be incorporated into the underlying logic of the software so that adherence to policy, process, and procedure can be monitored or enforced.

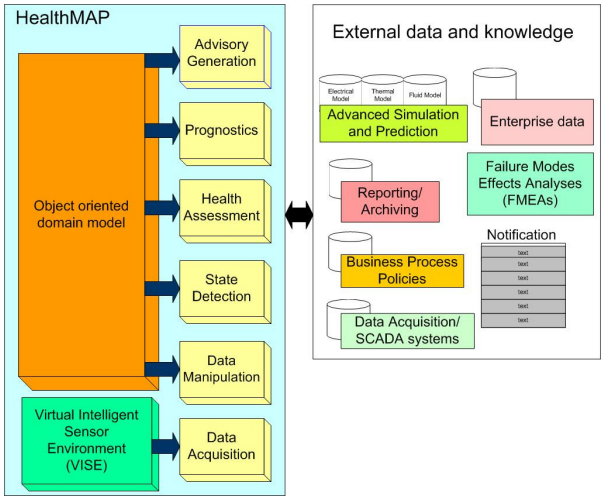

Figure 6: Typical HealthMAP™ Reasoner Architecture

HealthMAP™ also provides standards based external interfaces that provide prospective users with capabilities for data entry and display - typically accomplished through some graphical user interface

(GUI). Since most facilities already possess a number of systems with existing GUIs, HealthMAP™ has been designed to support a number of industry standard communications protocols. For example, the platform supports XML-based communication for generating web portals and displays that can be used to provide remote visibility into real-time system and equipment status. In this way, appropriate high-level status, pertinent historical data, and information regarding system availability can all be aggregated and made available for analysis by remote subject matter experts and executive-level responsible parties.

6 CONCLUSION

The importance of real-time system health monitoring for mission critical systems has increased as the users of these systems demand improved operational availability, greater reliability, increased safety, and reduced cost. The general goal of such PHM applications is to aid in the timely detection and/or prediction of failures that might result in increased safety hazards, unscheduled shutdowns, equipment and personnel casualties, negative environmental impacts, loss in production, or loss of revenue.

Defining requirements for PHM systems, however, has always been a challenge. Huge improvements in cost, schedule, and customer satisfaction can be realized by applying the concepts of RCM to the specification of PHM system requirements. In particular, with the application of the right set of tools and technologies, significant savings can be achieved for PHM development by reusing the domain knowledge developed during design analysis at the earliest stages of system conceptual design. We recognize, however, that integrating PHM system design considerations into the overall system design process requires a cultural shift across all organizations involved. While the benefits of such an approach are significant to both PHM system design and the overall system design, in practice a silo-based failure to communicate across disciplines is a significant roadblock to progress.

The implementation of integrated PHM systems is greatly facilitated by advanced reasoning systems that incorporate various advanced technologies. Tools that readily incorporate these technologies and provide access to high level reasoning capability are essential to delivering powerful PHM systems. While there are many effective tools and technologies developed by academia and industry for developing PHM systems or performing system design analysis, tools that support integrated system design and analysis and the development of run-time PHM are limited and constitute significant opportunities for PHM system designers and researchers. When coupled with model-based reasoning capability that incorporates the various required PHM system layers, we believe that full featured Prognostics and Health Management systems can be implemented faster and more efficiently.

ACKNOWLEDGMENT

The authors thank Scott Forney, Tony Kopacz, Carmelo Rodriguez, and Meera Venkatesh of General Atomics for their ongoing support of this technology. The authors also thank the General Atomics Intelligent Systems team, including Bruce Trumbo, Robert Gross, Kim Wilkins, Dave Vasil, and Tom Gogg, Jimmy Sonan, and Eduardo Suarez. The authors would also like to acknowledge the efforts of the NASA Stennis Space Center ISHM team, including Dr. Fernando Figueroa, Jonathan Morris, Harvey Smith, Mark Turowski, and Dr. John Schmalzel.

REFERENCES

(Ammerman, 1998) M. Ammerman. *The Root Cause Analysis Handbook.*. New York, NY.: Productivity Press, 1998.

(Brignolo *et al.,* 2001) R. Brignolo, F. Cascio, L. Console, P. Dague, P. Dubois, O. Dressler, et al., "Integration of Design and Diagnosis into a Common Process". *Electronic Systems for Vehicles,* Duesseldorf: VDI Verlag, 2001, pp. 53-73.

(Butcher, 2000) S.W. Butcher, *Assessment of Condition-Based Maintenance in the Department of Defense*, Logistics Management Institute, McLean, VA 2000.

(Elsayed, 1996) E. Elsayed. *Reliability Engineering.* Reading, PA: Addison Wesley, 1996.

(Figueroa *et al.,* 2006) F. Figueroa, R. Holland, J. Schmalzel, and D. Duncabage, "Integrated System Health Management (ISHM): Systematic Capability Implementation," *2006 IEEE Sensors Applications Symposium,* Houston, TX, April 2006

(Kapadia, 2003) Kapadia, R. "SymCure: A model-based approach for fault management with causal directed graphs", *Proc. Of the 16th Intl. Conf. IEA/AIE-03*, pp. 582-591, Loughborough, UK

(Kapadia *et al.,* 2007) R. Kapadia, G. Stanley, and M. Walker, "Real World Model-based Fault Management". *Proc. Of the 18th Intl. Workshop on Principles of Diagnosis,* Nashville, TN May 2007

(Kapadia *et al.,* 2009) R. Kapadia, R. Gross, M. Walker, M. Venkatesh. "Health Monitoring Assessment and Prognostics (HealthMAP™) for

Advanced Arresting Gear System. *Submitted to the 2nd Annual PHM Conference*, 2009.

(Kurtoglu *et al.*, 2008) T. Kurtoglu, S. Johnson, E. Barszcz, J. Johnson, P. Robinson. "Integrating System Health Management into the Early Design of Aerospace Systems Using Functional Fault Analysis". PHM'08, Denver, CO, 2008.

(Levitt, 2003) J. Levitt, *Preventive and Predictive Maintenance*. New York, NY: Industrial Press, 2003, pp. 123–135.

(DoD, 1980) *Procedures for Performing a Failure Mode, Effects, and Criticality Analysis* (Military Standard), MIL-STD-1629A, Washington D.C.: Department of Defense, 1980.

(Moubray, 1997) J. Moubray. *Reliability Centered Maintenance, Second Edition.* New York, NY: Industrial Press, 1997, pp. 123–135.

(NSSC, 2007) *Reliability-Centered Maintenance Handbook*, Washington D.C.: Naval Sea Systems Command, 2007.

(Okes, 2009) D. Okes. *Root Cause Analysis*. Milwaukee, WS: ASQ Quality Press, 2009

(ONRR, 1981) *NUREG-0492 Fault Tree Handbook,* , Washington D.C.: U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research,1981.

(Papadopoulos, 2003) Y. Papadopoulos, *Model-based system monitoring and diagnosis of failures using state charts and fault trees*. Amsterdam: Elsevier Science; 2003

(Patterson-Hine *et al.*, 2005) A. Patterson-Hine, G. Aaseng, G. Biswas, S. Narashimhan, K. Pattipati. "A Review of Diagnostic Techniques for ISHM Applications". ISHM Forum 2005, Napa, CA

(Puri *et al.,* 2006) G. Puri, D. Boylan, R. Walter, M. Lebold. "Building an OSA-CBM (Open Systems Architecture for Condition-based Maintenance) System". Penn State – Air Force Research Labs, Nov 2006.

(Stamatis, 2003) D. H. Stamatis. *Failure Mode and Effect Analysis: FMEA from theory to execution.*. Milwaukee, WI: American Society for Quality, 2003.

(Stamatelatos, 2002) D. Stamatelatos. *Probabalistic Risk Assessment Procedures Guide for NASA Managers and Practitioners.* Washington D.C.: NASA, 2002.

(Schwabacher and Goebel, 2007) M. Schwabacher and K. Goebel. A Survey of Artificial Intelligence for Prognostics, in *Proceedings of AAAI Fall Symposium*, Arlington, VA, 2007.

(Schoeller *et al.,* 2007) M. Schoeller, M. Roemer, M. Derriso. "Embedded PHM and Reasoning Integration Across Key Aerospace Vehicle Systems". *Integrated Systems Health Management Conference*, Cincinnati, OH, Aug. 2007.

(Walker, 2007) M. Walker, "Model-based Reasoning Applications for Remote Intelligent Systems Health Management", *ASNE Intelligent Ships Symposium,* May 2007

(Walker *et al.,* 2007) M. Walker, R. Kapadia, B. Sammuli, and M. Venkatesh "A Model-based Reasoning Framework for Condition-based Maintenance and Distance Support", *ASNE Automation and Controls Symposium,* ASNE: Biloxi, MS, Dec. 2007

Mark G. Walker received his BSEE from Cal Poly University, Pomona (1990), and his MSCompEng from the University of Southern California, Los Angeles, CA (1994), where he specialized in machine intelligence. He has been working in applied artificial intelligence since 1989, and has co-authored three patents in the field. His work with HUMS and PHM began with BFGoodrich Aerospace, Vergennes, VT in 1996. He also spent six years as Senior Consulting Engineer for expert system manufacturer Gensym Corporation. He has been with General Atomics since 2004, where he is employed as Lead Engineer, Intelligent Systems. He resides with his family in Oceanside, California.

Ravi Kapadia has a Ph.D. in Artificial Intelligence from Vanderbilt University, Nashville, TN. His research is primarily focused on model-based reasoning and its applications to diagnosis and design. Over the past fourteen years, he has published several papers in these areas. He is the primary developer of SymCure, a commercially available model-based diagnostic expert systems platform. He is currently at General Atomics applying model-based diagnostic techniques for Prognostics and Health Management Systems.