NASA/CR–2011-217147

Vehicle Integrated Prognostic Reasoner (VIPR) 2010 Annual Final Report

George D. Hadden, Dinkar Mylaraswamy, and Craig Schimmel
Honeywell International Inc., Golden Valley, Minnesota

Gautam Biswas, Xenofon Koutsoukos, and Daniel Mack
Vanderbilt University, Nashiville, Tennessee

May 2011

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at 443-757-5803

- Phone the NASA STI Help Desk at 443-757-5802

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/CR–2011-217147

Vehicle Integrated Prognostic Reasoner (VIPR) 2010 Annual Final Report

George D. Hadden, Dinkar Mylaraswamy, and Craig Schimmel
Honeywell International Inc., Golden Valley, Minnesota

Gautam Biswas, Xenofon Koutsoukos, and Daniel Mack
Vanderbilt University, Nashville, Tennessee

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NNL09AD44T

May 2011

Acknowledgements

We would like to thank the Honeywell and Vanderbilt teams for conducting the work under this program. In particular, we wish to acknowledge Howie Weibold for his many contributions, Darryl Busch, who kept us technically on track during the proposal and many reviews, and Dave Kolbet, Randy Magnuson, and Ginger Shao for their technical contributions to diagnostic, prognostic, and fusion algorithms. We would also like to thanks Hal Voges for his technical feedback and support, Craig Goodrich for helping us with all contract related issues, and Savit Boyd for her patience and her lessons on program management.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Table of Contents

Table of Figures

Table of Tables

1 Executive Summary

Honeywell's Central Maintenance Computer Function (CMCF) and Aircraft Condition Monitoring Function (ACMF) represent the state-of-the art in integrated vehicle health management (IVHM). These Honeywell products and technologies are purchased by airframe manufacturers such as Boeing, Bombardier, and Dassault and deployed on their aircraft. Underlying these technologies is a fault propagation modeling system that provides nose-to-tail coverage and root cause diagnostics. The Vehicle Integrated Prognostic Reasoner (VIPR) extends this technology to *interpret* evidence generated by advanced *diagnostic* and *prognostic monitors* provided by component suppliers to detect, isolate, and predict adverse events that affect flight safety.

VIPR brings together advances in subsystem health monitoring from several ongoing NASA, U.S. Army, and AFRL programs. This experience has given our team a unique insight to characterize heterogeneous and uncertain forms of evidence generated from these subsystem monitors and a reasoning system to correctly interpret them. We defined the data structures and algebraic operators for this interpretation.

In year one of VIPR, we laid the technical foundations for this next generation vehicle level reasoner that can be adapted to a variety of user requirements and deployed within aircraft computational constraints.

Significant accomplishments during the first year of the VIPR program include:

1. A basic three-tiered framework has been designed and illustrated through an animated ConOps demonstration (simulator).
2. Technical risks were mitigated through a comprehensive simulation of user requirements, animated ConOps, and architecture flow.
3. Deployment risks were mitigated through the extension of existing state-of-the-art diagnostic systems extended to handle heterogeneous evidence and provide prognostic conclusions.

We believe there are two important elements for moving the underlying technology into products expeditiously. First, it must address some of the safety gaps that exist today or user needs. Second, there must exist a pathway for realizing VIPR as cost-effective extensions to existing aircraft hardware and software. Using the ASIAS database, we identified four events as our demonstration scenarios. By its very definition, the current vehicle level reasoner was not able to detect the underlying fault event in all four situations and hence resulted as safety incidents. As part of our demonstration, we scripted how VIPR with its prognostic ability and advanced reasoning capability can not only detect these events accurately, but also allow sufficient time for the flight and maintenance crew to react and avoid the safety escalation. An animated concept of operations allowed various users to visualize how the VIPR system can address their unique needs.

Further, using an expanded set of ARINC 624 encoded messages, we also demonstrated how the VIPR can be realized as extensions to the existing Aircraft Condition Monitoring Function and the onboard Diagnostic Reasoner. Our prognostic reasoning formulation reuses existing diagnostic (fault propagation models) to a very large extent. Abstraction of evidence generation (monitors) provides a clear and practical way for 3rd parties to embed their knowledge and thus provide VIPR enriched information for vehicle level interpretation and reasoning. Within the software-based emulator environment we were also able to demonstrate how various advanced reasoning functions can be distributed to accommodate available aircraft computation resources. This is a first step for community acceptance and helps us to move the VIPR technology into products expeditiously.

2 Introduction and Background

An important challenge facing aviation safety today is safeguarding against system and component failures and malfunctions. Faults can arise in one or more aircraft subsystem; their effects in one system may propagate to other subsystems, and faults may interact. The primary function of a vehicle level reasoner is to detect faults and failures at the aircraft level, enable isolation of these faults, and estimate remaining useful life. All these functions are aimed at meeting the goal of automated mitigation and increasing aviation safety.

Consider characteristics of some typical faults arising in some subsystems within an aircraft:

1. [propulsion] Turbine blade erosion. This erosion is a natural part of turbine aging and wearing of the protective coating due to microscopic carbon particles exiting the combustion chamber. As the erosion progresses over time, it starts to affect the ability of the turbine to extract mechanical energy from the hot expanding gases. Eventually this fault manifests itself as increase in fuel flow and *gradual degradation* of engine performance.
2. [avionics/software] Loose wire harness connectors. As connector pins corrode, they make intermediate contact. The corresponding software module that receives this signal registers a series of *intermittent open circuit* faults. Eventually this corrosion progresses to a point which results in an open-circuit failure. Bad data from this channel corrupts the navigation software and causes a memory overflow instantaneously.
3. [airframe] Actuator stiction. A sticking actuator changes the dynamic response of a control loop. The feedback action provides some degree of resilience making this problem difficult to detect. But it does steadily decreasing the control loop’s ability to meet setpoint commands. Eventually, the stiction progresses to a point where the actuator will become non-responsive.
4. [software] This scenario describes a fast progression fault in which the incoming navigation data corrupts the guidance software (see ATSB Investigation report 200503722), which then leads to an incorrect solution. The auto-pilot intervenes and over compensates using the engine thrust. This causes high temperature and high speed events in the engine, leading to cascading problems in the generators and secondary power distribution system. Several auxiliary electronics modules react to the power glitch.

Broadly speaking the VLRS needs to address scenarios wherein (1) the underlying fault progresses both in time and severity and (2) the effects of a fault are felt throughout the aircraft and its operations. More specifically:

1. Faults whose severity increases with time. These can be further categorized based on the time constant of this evolution such as incipient, slow progression or fast progression.
2. Binary repeating faults whose repetition increases with time. These can be further categorized based on the time interval between repeats such as constant or increasing.

3. Faults whose effects spread throughout the aircraft with time. These can be further categorized based on the size of this influence such as localized (self contained) or widespread.

Honeywell's Aircraft Diagnostic and Maintenance System (ADMS) that reasons using a fault propagation system model is a state of art in vehicle level *diagnostic* reasoning. On the other hand, the Joint Strike Fighter (JSF) Prognostics Health Management System represents a state of the art in *generating prognostic indicators* at the subsystem level. Interpretation of these prognostic indicators is important to meet the goal of automated mitigation and increasing aviation safety. Our primary research is to extend vehicle level reasoning by incorporating these prognostic indicators and design, implement and demonstrate a vehicle level integrated prognostic reasoner. We call this VIPR (**V**ehicle **I**ntegrated **P**rognostic **R**easoner).

Evidence provided by a prognostic indicator needs to be interpreted differently from an evidence provided by a binary on/off indicator. Mathematical characterization of these heterogeneous forms of evidence is an important part of VIPR design. The reasoning within VIPR needs to address a multitude of timescales involved in the evidence as well the coverage of aircraft subsystems and their interactions. Decomposing this reasoning into small inferencing steps is necessary to manage complexity. With the introduction of new aircraft and retrofit of current platforms, a clear articulation of the architecture options is more important than the underlying reasoning technologies.

Data mining and machine learning techniques provide the primary mechanism for characterizing interactions between components, subsystems, and potential causal chains of adverse events that impact safety. The underlying algorithms support VIPR program goals by (1) establishing the parametric relationship (probabilities, coefficients, etc.) associated with various entities in VIPR fault propagation system model and (2) discovering new relationships from operational data. Often, the limiting factor is availability of realistic data. The data necessary for this activity need to retain statistical richness while maintaining privacy and proprietary restrictions.

While designing VIPR presents unique research challenges, the safety benefits from a vehicle level reasoner can only be realized from its acceptance within the aviation community. There are two important elements here: (1) articulation of user requirements and (2) demonstrating how VIPR detects and predicts faults and failures before they escalate to flight safety incidents.

Developing a next generation vehicle level reasoner embodies several risks.

1. Inferencing operators and data model design for prognostic reasoning present technical risk;
2. Non-availability of real data present credibility risk;
3. Improper capture of user requirements presents practical realization risk;
4. Lack of clarity in the "end state" and safety impact constitutes adoption risk.

It is important to addresses all these risks before "building the VIPR solution". This risk reduction step was the primary objective of our effort and this report summarizes our process and delivered artifacts that culminate in a set of recommendations and future tasks for realizing a practical VIPR with high degree of success. We begin with an overview of VIPR in section 3. Progress and deviations made from the VIPR concepts described in the original proposal is described in section 5. Documents, artifacts and demonstrations that contribute towards risk reduction are described in section 6. We conclude this document by summarizing future steps for designing, implementing and demonstrating a successful VIPR.

3 Objectives/Approach

Objectives for the VIPR program flowed from the overarching NASA objective of achieving a higher level of aircraft safety through an embedded Vehicle Level Reasoning System. Our year one work included defining the architecture and communication protocols and establishing user requirements. Based on these and a set of scenarios defined in our ConOps document, we designed and implemented a demonstration using Honeywell's SMARTlab simulation facilities. In addition to demonstrating the communication pathways and the three-tiered health management architecture, scripted scenarios show VIPR's ability to detect adverse events before they escalate as safety incidents. This demonstration testbed is designed so that future work can add reasoning software for prognostics and diagnostics and later, actual aircraft hardware.

The year one objectives included those mentioned above as well as making available to the IVHM community a large set of data acquired from the Mesaba BAe RJ fleet and were realized through the performance of the following major tasks:

- Architecture Recommendations. Produce and document recommendations for the VLRS architecture addressing areas such as data transfer protocols, speeds, and communications requirements for airframe, propulsion, aircraft, and software subsystems. The recommendations, requirements, and associated metrics should be based on the needs of the user community.

- Information Protocol. Develop a health management information protocol that includes requirements for the information and formats needed to be passed through all levels of the VLRS.

- Concept of Operation. Provide a concept of operations of the VLRS including a study of the trade-space between complexity, accuracy, cost, and impact on aviation safety. The trade space between the numerous (and sometimes conflicting) user requirements and the customer's desire to minimize cost should be clearly documented.

- User Requirements. Develop a comprehensive set of user-requirements for Condition Based Maintenance and the application of the VLRS to enable appropriate predictive maintenance based on a fleet management perspective. Document in a NASA Technical Manuscript or other peer-reviewed publication.

- Metrics Recommendations. Provide recommendations regarding appropriate metrics for CBM in the context of all of the subsystems mentioned above and discuss how the proposed VLRS addresses those metrics. Document in a NASA Technical Manuscript or other peer-reviewed publication.

- Tools and Technology Concept of Operations. Provide a concept of operations of the VLRS tools and technology, describing the potential cost-benefit tradeoffs in terms of CBM for a real-world aircraft that can be enabled by the VLRS. Requirements and cost benefit analysis should be documented with respect to user requirements that they are supporting or trading off (logistics, maintenance, flight, fleet management, training, etc.) Document in a NASA Technical Manuscript or other peer-reviewed publication.

- Demonstration. Demonstrate the proposed concept of operation in a software simulation for a subset of seeded faults (selected from the Table 2 Adverse Events Table IVHM Tech Plan) in a vehicle configuration consisting of at least three different subsystems.

Our recommendation for future work includes the migration of the demonstration system to the use of diagnostic and prognostic reasoning software. We recommend this software be augmented and tuned using data mined from the Mesaba data and other sources. Eventually, hardware should be inserted into the VIPR system in order to demonstrate its capabilities on real world problems. Metrics should be defined and applied to the VIPR system to discover to what extent (if any) it is superior to existing systems.

4 VIPR Overview

Similar to a CMC, VIPR has several users. In year one, we focused on the flight crew as primary consumers of VIPR outputs. The second set of users are line maintainers and repair depot maintainers. The third set of users includes the systems integrators responsible for installing and maintaining VIPR. Flight crew requirements include recognition of conditions that may cause an adverse event, mapping it to functional effects, and verifying that the designed contingency (if any) is working properly. Systems integrator requirements include clear separation of evidence generation (called *monitors* and supplied by component manufacturers), aircraft configuration, and a common code base for minimizing certification costs as well as a hierarchical architecture that can be deployed within aircraft communication and computation constraints.

It is not surprising that these user requirements imply a need for different views of the situation. VIPR solves these problems by starting with a well-defined separation between evidence generation, a reference model that encodes aircraft specific configuration data, and a generic platform agnostic DP (diagnostic/prognostic) reasoner to provide a common code base that allows for one-time certification. Recommendations that center on allowing LRUs to interface with VIPR while allowing their manufacturers to maintain control of their intellectual property were documented as a deliverable in year one.

To address the spectrum of events that adversely affect aviation safety, the underlying reasoning algorithm must work on enriched evidence generated by proprietary monitor providers. While VIPR does not care about the internal proprietary knowledge, an abstraction into simple, multivariate, multiclass, and prognostic monitors allows VIPR to formalize the uncertainty and heterogeneity associated with the collected evidence. Motivated by our work on the Army's Future Combat Systems (FCS) Platform Soldier-Mission Readiness System (PS-MRS) program, we defined *fault condition* as a fundamental data structure within the reasoning process. The persistent set of these data structures maintained within the VIPR software along with their attributes establishes the prevailing fault hypotheses (for a maintainer), functional effects (for flight crew), and monitors of interest (for active fault isolation and data capture). This definition is accompanied with a set of operators for creating, merging, splitting, resolving, and closing these fault conditions as new evidence arrives.

The diagnostic and prognostic processes reduce to a set of configurable meta rules that applies these operators whenever a piece of evidence is generated. Applying these operators requires computational resources. However, unlike the CMC, where all these computations are done at a central location, VIPR includes a hierarchical tiered and distributed architecture. This enables subsets of these operations to be applied at the computationally most suitable location within the aircraft to meet the timeliness need of detecting fast adverse events. The need for information and data passing is met by defining message passing protocols based on ARINC 624 encoding.

VIPR embodies new concepts and new technologies. Validating these definitions early on is not only important to increasing the likelihood of technical success, but also important for early adoption within the community. The SMART (Simulation and Modeling for Acquisition, Requirements, and Training) process (see Section 6.2) emphasizes intuitive visualization, ensuring that customers "see" the VIPR architecture design elements early and often. We concluded year one with a series of animated concepts of operations that clearly highlighted various design concepts within VIPR. Events pertaining to the Mesaba airline fleet recorded in Aviation Safety Information Analysis and Sharing (ASIAS) database provided us scenarios for visualizing the VIPR design elements and benefits with respect to safety and maintenance.

Analyzing actual flight data provides the right level of validation for measuring the accuracy of VIPR. Our development of an anonymizer to remove proprietary encoding allows us to distribute the Mesaba data for analysis and data mining. Future work involves quantification of the reasoner accuracy and the VIPR design trade space using this data and metrics. Section 6.4 presents our recommendations for how data mining can be used to enhance the VIPR reference model.

5 Progress Summary

The VIPR proposal was built on seven key concepts (Table 1). Three of these concepts (fault condition construct and operations, system reference model, monitor and evidence abstraction) were related to the reasoning algorithm. The SMART process concept allows visualization of the VIPR design elements as they evolve not only for trade studies but also for early adoption within the community. Mesaba data and data mining concepts provided the necessary tools to continually refine the reasoner and discover new knowledge. Activities in year one fleshed out the design definitions and requirements and shortcomings. The animated ConOps demo (step 2 in the SMART process) not only helped us to communicate the design through visualization, but also allowed us to zero-in on gaps and make course corrections. The VIPR design now stands on a solid ground and we can say that we will be entering the implementation phase (year two) with high degree of confidence to meet all our proposed goals. Table 1 summarizes the key concepts, progress to data and course corrections from year one.

Table 1. Accomplishment Summary and Course corrections with respect to proposal concepts

Proposal Elements	Progress to date	Course Correction/Lessons
Fault condition construct and operations	Definitions complete. Salient features captured in the VLRS Concept of Operations document, **CDRL 4.1.04**. On track to implement them within the SMARTlab simulator	We expanded our original evidence set to include human provided evidence. Correspondingly we expanded the operations on fault conditions to interpret these TWO forms of evidence.
System Reference Model	Definitions and requirements capture complete. Described in User Requirements, **CDRL 4.1.05**. On track to instantiate a reference model for the propulsion, bleed, avionics and aircraft actuator subsystems.	We discovered the need to extend the System reference model to include "data of interest" elements. During configuration time, this allows the VIPR installer to specify the sensors used by monitors. This information allows the reasoner to capture data with the onset of a primary evidence, and plays an important role in sensor fault isolation.
Monitor and evidence abstraction	We defined six forms of evidence heterogeneity. We also defined an abstraction for capturing uncertainty associated with these monitors without exposing proprietary knowledge to vehicle level reasoning complete. Captured in Architecture Recommendations, **CDRL 4.1.02**.	The monitor abstraction proposed originally could not handle evidence provided by humans. We extended the abstraction to include TWO forms of human monitors: *loss of function* and *loss of asset*.
Tiered & distributed architecture	We defined messaging protocols to support distributed reasoning. Implementation of these message passing protocols within the simulator is complete. The protocol is defined in **CDRL 4.1.03**.	We discovered that ARINC 624 has proven precedence on commercial aircrafts to support diagnostic messages. We decided to adopt this protocol and expanded it to include information content requirements derived from the AFRL program ISHMAD [Jambor].
SMART Process	Demonstrated the following steps of the process--Animated ConOps, architecture flow-- for four scenarios spanning five aircraft subsystems (propulsion, bleed, avionics, actuators, and software). Summary of these scenarios are described in **CDRL**	No significant course corrections.

	4.1.04. User requirements were captured in a document, **CDRL 4.1.06**.	
Mesaba Fleet data	*We identified ten incidents recorded in the ASIAS database and relevant ACMF data from the Mesaba fleet. The ACMF data provides 1—16Hz aircraft parameters spanning at least 40 flights before and after the events.* We completed the data anonymizer that allows us to make this data available to the VIPR program and NASA.	The Mesaba archive does not include data specifically captured for supporting software health monitoring. Our recommendation is to simulate these faults based on historical scenarios.

6 Year One Deliverables

6.1 User Requirements

Figure 1 shows the system boundary diagram for VIPR. The figure identifies various users that will interact with a vehicle level reasoner such as VIPR. The users include both consumers of information as well as providers of information.

Primary users of VIPR information considered in this report are shown using solid circles. These include: (1) the flight crew that is operating the aircraft and their requirements to detect adverse events and mitigate effects of such events to increase aviation safety, (2) the VIPR installer who is responsible for assembling and installing the VIPR system for the aircraft, and (3) the VIPR maintainer who is responsible for performance evaluation and continual upgrades to reflect changing aircraft configurations.

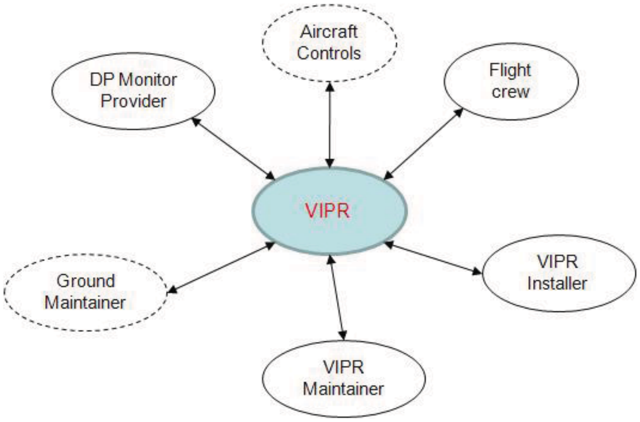
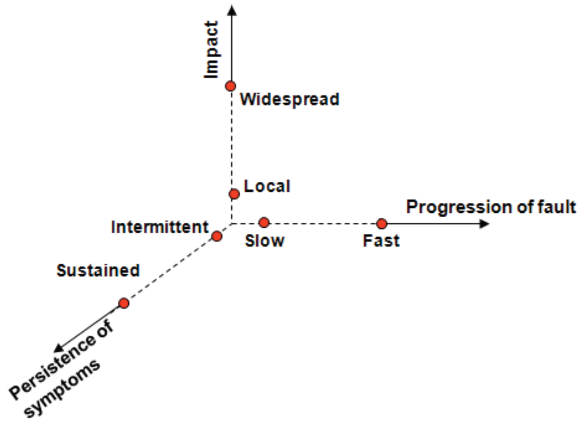


Figure 1. System Boundary Diagram

Secondary users include (1) providers of diagnostic and prognostic monitors (e.g. LRU manufacturers), (2) the ground maintainer responsible for performing inspections and repair actions, and (3) the aircraft control systems for semi-automatic and automatic mitigation in response to detection of adverse events.

On this task, we developed a novel mechanism for describing adverse events in the vehicle.



See Figure 2. This mechanism consists of a cube with three mutually orthogonal axes labeled time evolution (with extremes labeled fast and slow), impact propagation (with extremes labeled localized and widespread), and symptom persistence (with axes labeled intermittent and constant). We believe points in this space correspond well to events that VIPR needs to address to increase aviation safety.

Figure 2. Adverse Events Cube for gathering flight crew requirements

Table 2 summarizes key flight crew requirements.

Table 2. Flight Crew Requirements Summary

	Event Type	Top Level requirements (Flight crew)
Time Evolution	Slow	1. Less important. 2. Important, if and only if it will affect the current flight.
	Fast	1. Very important. Early detection of incipient conditions. 2. Quickly identify mitigation (could be automatic control) actions
Impact Propagation	Localized	1. Less important. 2. Confirm and monitor if redundancy is working as designed
	Widespread	1. Minimize information overload to avoid confusion. 2. Suppress information presentation, do not remove the evidence.
Symptom Persistence	Constant	1. Reduce false alarms. 2. Minimize size of Ambiguity group and rank order.
	Intermittent	1. Accurate detection and establish that intermittency is true. 2. Identifying a root cause may not be important

Key requirements for a VIPR installer are summarized in Table 3.

Table 3. VIPR Installer Requirements Summary

	Top Level requirements (VIPR Installer)
Scalability	1. Separate the reasoning algorithms from aircraft specific configurations. 2. A common code base is easy to validate and makes is easier to certify. 3. Finite set of operations, each of which is bounded computationally.

Deployment	1. Reasoning function needs to fit on available onboard hardware. 2. Support LRU's that do not have computational resources for generating monitors. 3. VIPR should work within the intellectual property boundaries of a monitor provider. 4. Unambiguous definition of monitor types to avoid misinterpretation.
Accuracy	1. Ability to handle multiple timescales. Timestamp of evidence is important. 2. Must include 'states' (necessary and sufficient description) that can be archived and used as initial conditions for analysis across successive flights. 3. States are tracked using probabilities and well-defined 'update' operations 4. Capable of proposing and working with multiple fault hypotheses.

A version of this report deliverable was also published as part of an invited paper in the AIAA Infotech@Aerospace 2010 Conference entitled "Architectures for Integrated Vehicle Health Management" by Tim Felke, George D. Hadden, Dave Miller, and Dinkar Mylaraswamy.

6.2 Honeywell 7-Step SMART Process

VIPR embodies new concepts and technologies that integrate and reason about data captured from multiple subsystems in order to detect a potential adverse event, diagnose its cause, and predict the effect of that event on the remaining useful life of the vehicle. Validating these concepts early on is not only important to increase the likelihood of technical success, but also important for early adoption within the community. The SMART (Simulation and Modeling for Acquisition, Requirements, and Training) process developed by the US Army is a systems lifecycle modeling environment that

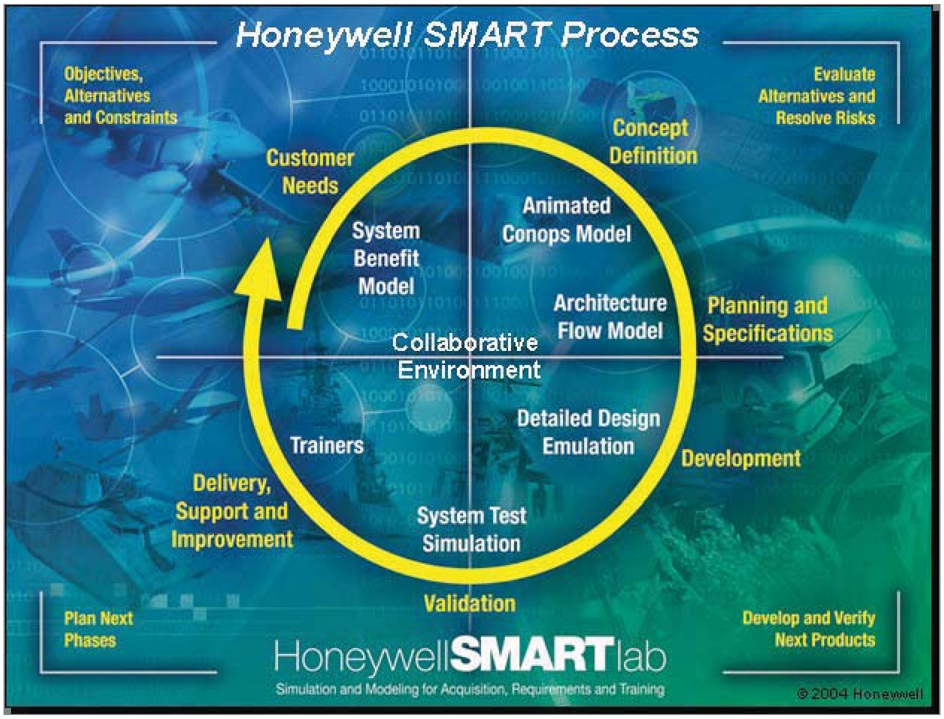

Honeywell SMART Process
Objectives, Alternatives and Constraints
Evaluate Alternatives and Resolve Risks
Customer Needs
Concept Definition
System Benefit Model
Animated Conops Model
Architecture Flow Model
Planning and Specifications
Collaborative Environment
Detailed Design Emulation
Development
Trainers
Delivery, Support and Improvement
System Test Simulation
Validation
Plan Next Phases
Develop and Verify Next Products
HoneywellSMARTlab
Simulation and Modeling for Acquisition, Requirements and Training
© 2004 Honeywell

Figure 3. Honeywell 7-Step SMART Process

emphasizes intuitive visualization, ensuring that customer “sees” the architecture design elements early and often.

The 7 steps are, as shown in Figure 3:

1. System Benefits Model: An early lifecycle, low fidelity model that demonstrates the utility of a system or process. This step often explores the cost to benefits tradeoffs.
2. Animated Concept of Operations (CONOPS) Model: A visual, animated model or prototype that illustrates the system operation. This model is used to confirm the project approach and customer expectations and acts as a concrete representation of the system requirements.
3. Architecture Flow Model: An interactive model that defines the interactions and information flow among system components. This model defines the roles and interfaces between subsystems.
4. Detailed Design Emulation: A high fidelity model that represents the final system. It provides validation of algorithms and system operation prior to major purchases or development activities.
5. Integration Testbed: A simulated environment capable of interfacing with real and simulated subsystems for integration of real assets as development matures.
6. System Test Simulation: A combination of real and simulated components that provides a realistic test environment without the risk or cost of a live test.
7. Training Systems: High fidelity models that may include real or simulated components, used to train operators in a controlled environment. (Note: Step 7 is not currently within VIPR’s scope.)

The result of each step flows into the next, so that each step expands on and refines the models of previous steps. Each step of the process can be employed iteratively and recursively throughout the development cycle. For example, if the animated CONOPS model exposes requirements issues, the model would be iteratively refined until the issues are resolved. Once the model accurately reflects the system, the requirements are updated to match the model and the design process continues.

6.3 Architecture

Architecture recommendations fell into four categories: Modular Solution, System Integration, Reasoning Algorithms, and Evaluation Metrics.

An important recommendation in the Modular Solution category is to use the ISO-13374 (OSACBM - Open Systems Architecture for Condition Based Maintenance) functional decomposition as a baseline for defining the VIPR processing blocks. An additional recommendation from this section is to base VIPR’s internal communication protocol on that developed for the AFRL ISHMAD (Air Force Research Lab Integrated System Health Management Architecture Design) updated to be consistent with the ARINC 624 standard (see Message Protocols section).

The most important of the System Integration recommendations is that the VIPR architecture be built on a three-layer hierarchy (see Figure 4). These layers comprise the LRU Health Manager at the lowest level, the Area Health Manager (concerned with interactions

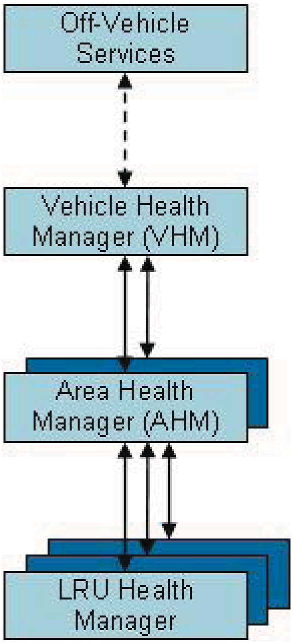


Figure 4. The VIPR Three-Layer Hierarchy

within subsystems, e.g. the engine), and the Vehicle Health Manager (which allows VIPR to draw conclusions based on events in separate parts of the vehicle). Off-vehicle services are beyond VIPR's current scope, however we recognize the importance of these services and will avoid design decisions that make this capability difficult to add.

Other System Integration recommendations center on allowing LRUs to interface with VIPR while allowing their manufacturers to maintain control of their intellectual property. This is done by defining an interface to VIPR using *monitors* to carry diagnostic and prognostic information from the LRUs. LRU internal operations need not be visible to VIPR. In this section, we also recommend that the communication protocols called for in the Modular Solution category be distributed as an open source library.

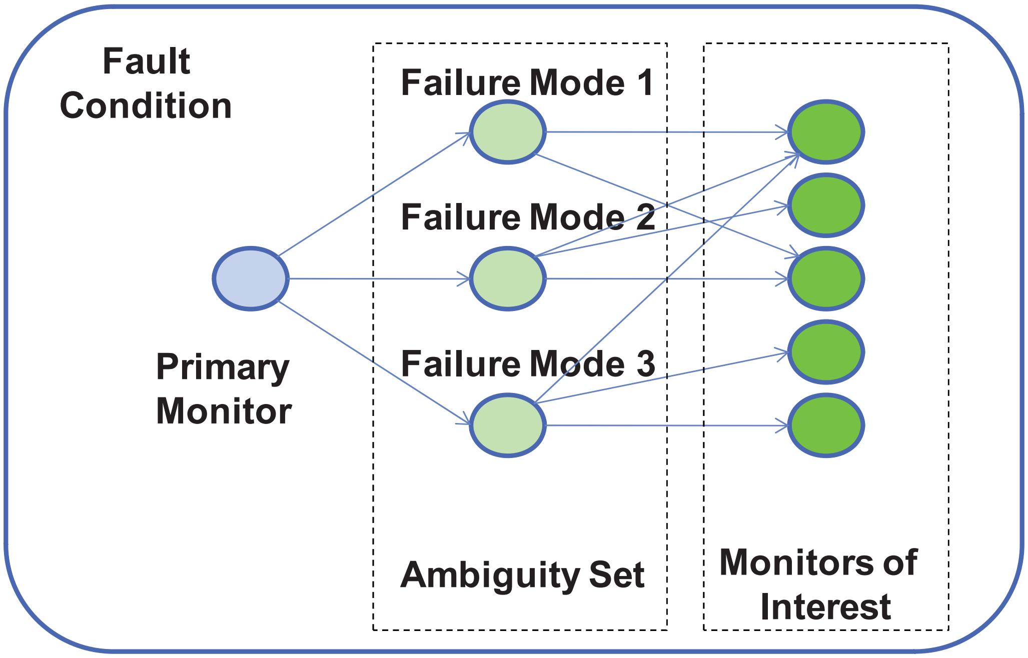


Figure 5. A Fault Condition

A key Algorithm recommendation is to use the *fault condition* as a fundamental data structure. Fault conditions contain a set of failure modes (called the ambiguity set), exactly one of which is assumed to be occurring. Fault Conditions also contain the set (called the "Monitors of Interest" of all monitors that might fire if any of the failure modes in the ambiguity set were to occur. Figure 5 shows this graphically. Multiple simultaneous faults can be diagnosed – and prognosed – using multiple fault conditions each of which maintains its own ambiguity set and a set of evidence to look for.

The Evaluation Metrics section defines six metrics: time to detection, detection accuracy, time to isolate, size of the ambiguity set, false alarms, and missed detections. Our recommendation from this section is to leverage previous Air Force work (as described above) as well as diagnostics and prognostics efforts on the Army's Future Combat Systems Platform Soldier Mission Readiness System (FCS PS-MRS).

The subsystems that VIPR addresses include Propulsion, Avionics, Airframe, and Software.

6.4 Reasoning Mechanisms – Tools and Technology Concept of Operations

The primary functional units of the VIPR architecture are: (1) the reference models that contain the information that the reasoning algorithms use to derive diagnostic and prognostic conclusions, (2) the message passing protocols (described more fully in the report deliverable for Task 4.1.03), and (3) the layered (LRU-, Area-, and Vehicle-level) diagnostic and prognostic reasoners. Details of the algorithms at the area level are described, as well as fusion algorithms that unify the results at the vehicle level. The approach to the two way interactions between the proprietary LRU monitors and the reasoners, i.e., bottom up information passing and top-down querying to refine diagnostic hypotheses is presented. In addition, the report discusses appropriate evidence combination schemes for representing and reasoning with uncertain data. This report then outlines how the new layered reasoners impact the Concept of Operations of aircraft health management systems and provides an example.

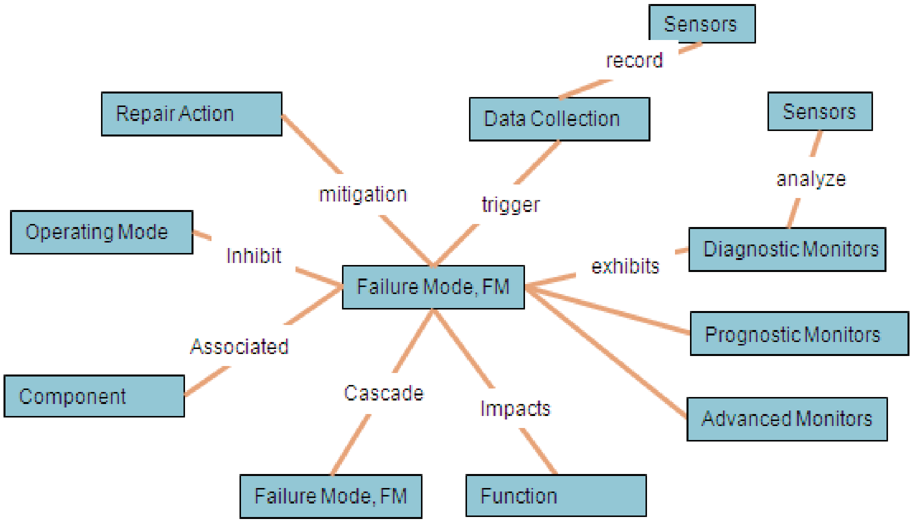


Figure 6. VIPR Reference Model Entities

Figure 6 Illustrates a number of entities and relationships captured in the VIPR Reference Model. Figure 7 shows how the various functional elements of VIPR map into the three-tiered architecture described above.

Schemes for continually improving the reasoning algorithms with operational field data are outlined in [Biswas, 6/2010] section 7, and recommendations are made for some of the considerations in preparation for a future VIPR data mining task.

With more precise knowledge of the fault condition structures and reference model, we recommend using Tree-Augmented Naïve Bayesian Network (TAN) structures to learn new relations rather than general causal discovery algorithms, such as TETRAD. TAN structures are interpretable, modifiable, and more easily derived. Combining TAN structures with local causal discovery algorithms provides the framework for continually improving the reasoning algorithms with operational field data.

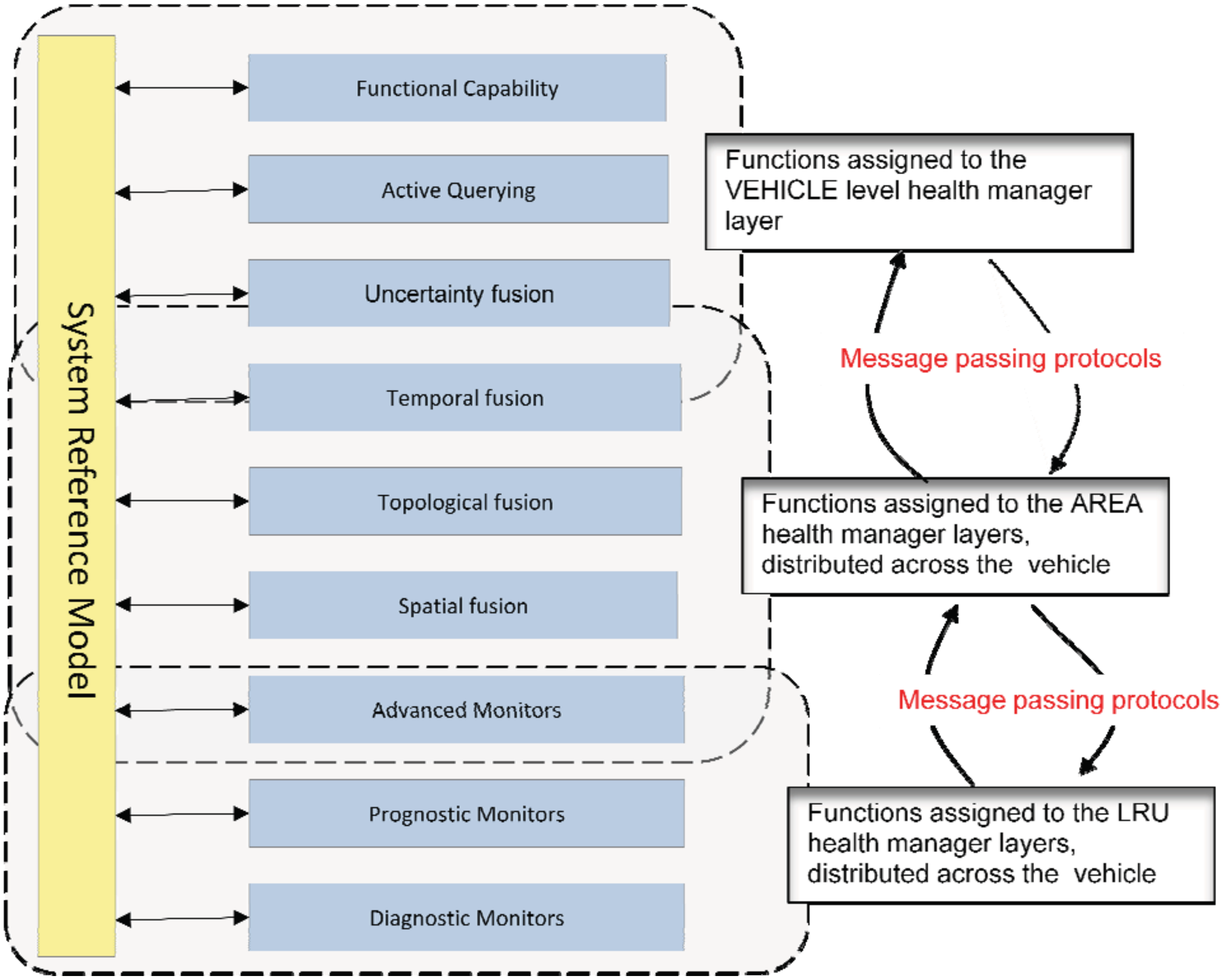

System Reference Model
Functional Capability
Active Querying
Uncertainty fusion
Temporal fusion
Topological fusion
Spatial fusion
Advanced Monitors
Prognostic Monitors
Diagnostic Monitors
Functions assigned to the VEHICLE level health manager layer
Message passing protocols
Functions assigned to the AREA health manager layers, distributed across the vehicle
Message passing protocols
Functions assigned to the LRU health manager layers, distributed across the vehicle

Figure 7. Functional view of VIPR

6.5 Message Protocols

VIPR operates largely by passing messages throughout its subsystems. We have based these messages on those developed for the AFRL ISHMAD program and on the ARINC 624 standard. VIPR contains seven basic types of message (Broadcast, Command, Event, Query, Command Response, Event Response, and Query Response) as well as a simulation specific message type used for demonstrating VIPR functionality. These messages are listed in Table 4.

Table 4. Message Types

Message Type	ARINC 624 equivalent	Description
Broadcast	Periodic Report	Broadcast messages are of interest to multiple elements and contain such information as flight phase and time.
Command	Command ACTION	Command messages to operate the vehicle are issued from VHM and maintenance crew. Acknowledgment is sent from receiver and often contains data response.
Event	Event REPORT	Anomalies are detected and sent to higher-level health managers as events. Messages contain originator, event type, time, location, analysis

		and supporting data. Includes Status, Capability, Maintenance, and Event Observe/Orient/Decide messages.
Query	Parameter GET	Query messages can request additional data.
Command Response	Command RESPONSE	Acknowledges the receipt of a command. Can include data confirming the results of the command.
Event Response	Event Ack	Acknowledges the receipt of an event message.
Query Response	Parameter STATUS	Provides the data requested by a Query message.
Sim Exec	-	*Simulation specific messages for demonstrating the VIPR functions.*

Each message has a standard (common) header, a specific message sub-header, data payload and a signature. The common header contains top level information such as the sender, destination, time, unique number, and message type. Putting this information in a common message header ensures that the messages can be delivered to their intended destination and interpreted correctly no matter what protocol is used to send them. The timestamp and message number fields also promote traceability of the messages. Most messages are further defined with a sub-header that provides additional information. The header and sub-header are encoded using ARINC 624 protocols. The maximum size for a single data payload is 64KB. However, multiple messages can be "chained together" to accommodate data greater than 64KB. Figure 8 shows the layout of a VIPR message.

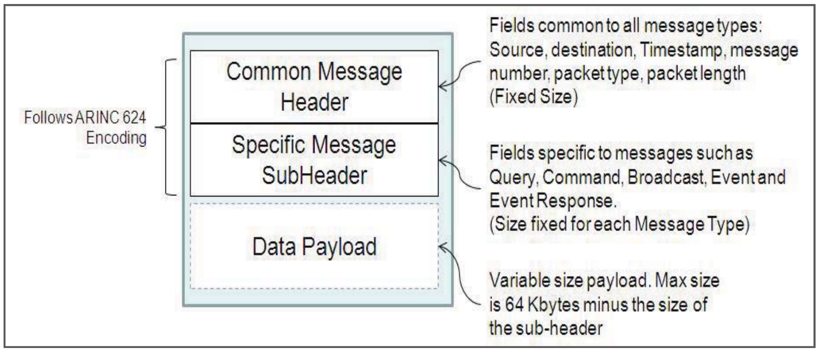

Follows ARINC 624 Encoding
Common Message Header
Specific Message SubHeader
Data Payload
Fields common to all message types: Source, destination, Timestamp, message number, packet type, packet length (Fixed Size)
Fields specific to messages such as Query, Command, Broadcast, Event and Event Response. (Size fixed for each Message Type)
Variable size payload. Max size is 64 Kbytes minus the size of the sub-header

Figure 8. Layout of a VIPR message

The chronology of events can be very important to the reasoning function. While a timestamp is included in every message header, we believe additional timestamp data may be needed for reasoning about fast progression faults. Therefore, an additional sampling time is included in each event sub-header to precisely define when that particular event occurred. However, VIPR does assume that timekeeping is well-synchronized across all subsystems. The broadcast messages (defined in this document) and the temporal fusion block (Figure 6) are intended to be a starting point to accomplish time synchronization. It is likely that the VIPR prognostic reasoner may be robust to handle small errors in this synchronization step. If not, the additional sampling time in the event message protocol can be used to experiment with more complex temporal fusion logic. This combination of an extensible protocol and a flexible architecture will allow us to synchronize time across subsystems event messages down to whatever resolution will be required to detect fast progression events such as software failures.

The report deliverable CDRL 4.1.03 contains detailed descriptions of the message protocol as well as a description if ARINC 624 encoding as it applies to VIPR.

6.6 ConOps and Scenarios

The Concept of Operations (ConOps) illustrates the VIPR architecture through a set of scenarios. In the report deliverable for this task we describe a separate scenario for each of the following aircraft systems: engines, flight actuators, and software. These scenarios are also illustrated in the VIPR Demonstration (Task 4.1.07). The scenarios describe the initial conditions prior to the occurrence of each fault, then use sequence diagrams to follow the fault through the diagnostic and prognostic functionality of VIPR and in some cases calculate the impact of the fault on aircraft functional availability. They are based on ideas stemming from reports in the Aviation Safety Information Analysis and Sharing (ASIAS) database, National Transportation Safety Board (NTSB) reports, etc., as well as observations we have documented through our flight data recorders. The scenarios cover a spectrum of Adverse Event Types listed in Table 2 of the NASA-IVHM Technical Plan.

The highlights of the four scenarios included in the demonstration are presented below.

1. Slow, progressive fault event with the fuel metering component of an engine. This scenario was based on the Mesaba airline incident which eventually led to an *in-flight engine-on-fire* alarm. Figure 9 shows the key information used by VIPR to diagnose this event.

   The scenario starts with relatively benign observation made by a monitor—the left engine has a slow start. This scenario illustrates VIPR ability to create a fault condition with several possible root causes from a relatively non-critical symptom. The fault condition is disambiguated as evidence emerges from successive flights, and active query mechanism within VIPR. The correct root cause is identified five to six flights before the *in-flight engine-on-fire event*.

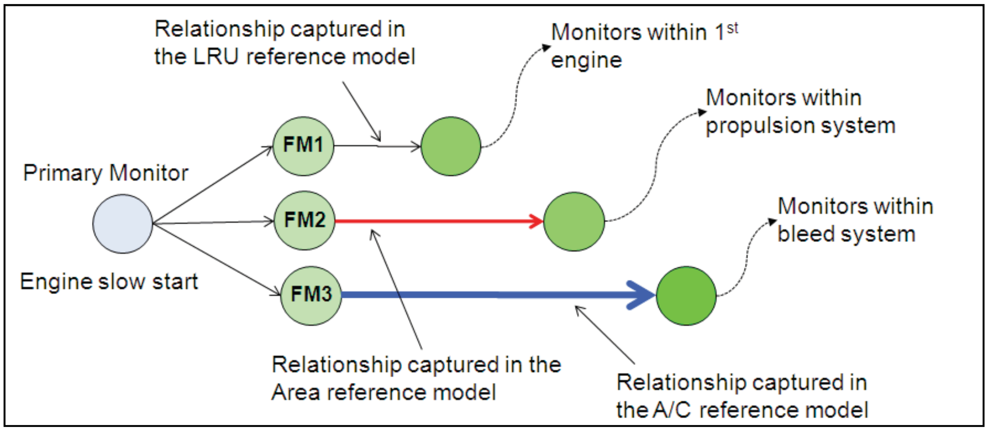

Relationship captured in the LRU reference model
Monitors within 1st engine
Monitors within propulsion system
Primary Monitor
FM1
FM2
FM3
Monitors within bleed system
Engine slow start
Relationship captured in the Area reference model
Relationship captured in the A/C reference model

Figure 9. VIPR ConOps for a slow, progressive event

As shown in See Figure 9, at the area-level, VIPR uses the fact that there are two engines and uses this information to compare the start times and eliminate common cause such as cold oil or fuel pump problems. At the vehicle-level, VIPR uses the physical connection between the engine and the bleed system to eliminate problems with the engine compressor and focus on the engine electrical components. At the LRU-level symptoms generated by various monitors are tracked over multiple flights and aggregated to increase the confidence level in the fuel metering unit.

2. Widespread symptom cascade fault event associated with the loss of an Air Data Inertial Reference Unit (ADIRU). This scenario was based by the Singapore airline (A330, 10/7/2008) incident, wherein a failed ADIRU caused multiple fault notifications and led *un-commanded pitch down events*.
   The scenario starts with flight controller switching from a primary channel to a secondary channel. The root cause is a bad ADIRU signal. Soon the failed ARIDU cascades as symptoms from the navigation and the ground Proximity sub-systems. VIPR follows the cascades chains to identify the root cause. Multiple symptoms generated from several connected subsystems are consolidated and *explained away* by root cause analysis. Once the fault is localized, VIPR uses symptom cascade relationship to identify a common root cause, namely the ADIRU bad values.

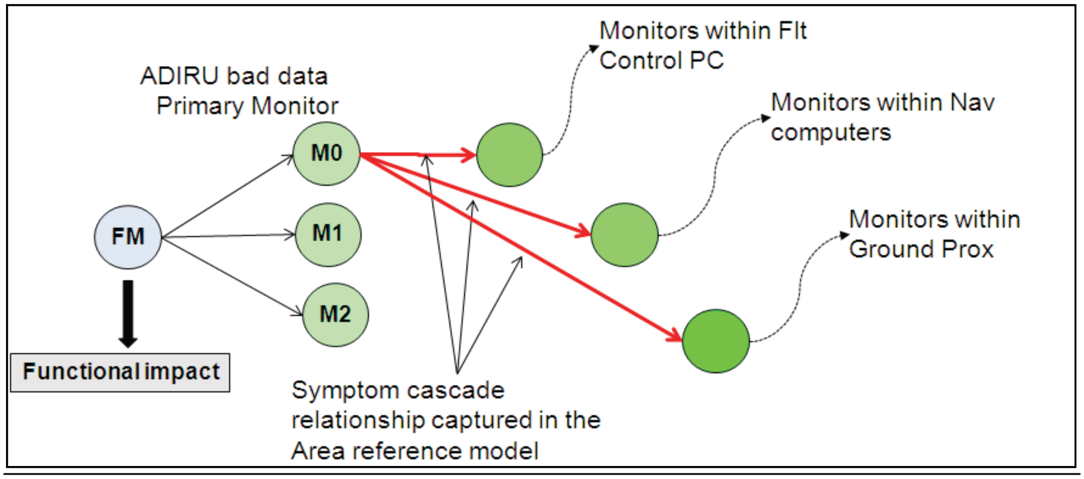

Monitors within Flt Control PC
ADIRU bad data Primary Monitor
Monitors within Nav computers
M0
FM
M1
Monitors within Ground Prox
M2
Functional impact
Symptom cascade relationship captured in the Area reference model

Figure 10. VIPR Conops for a wide-spread impact event

Throughput the evolution, VIPR manages the cascade and explains away various fault codes generated by subsystem that is connected to this ADIRU. Additional monitors from the Ground-Prox system exonerate the inertial reference (IR) subsystem, while indicting Air Data Unit (ADR). VIPR then proceeds to calculate the functional effects of this fault and informs the flight crew about alternative control laws that can prevent secondary effects such as un-commanded pitch-down events.

3. Sensor induced fault events, wherein a faulty sensor feeding triggers intermittent evidence. This scenario was based on a Mesaba airline incident wherein a inrange sensor fault caused intermittent loss of engine performance and eventually the flight crew returned back to the base after being airborne for 15-20 minutes.

An inrange sensor bias is extremely difficult to detect. An inrange bias does not cause a high-limit or low-limit exceedance and hence does not get detected by standard range-check algorithms. However, the numerical offset spoofed several diagnostic monitors. The result was the creation of several fault conditions – that indicated inlet fouling, compressor blade erosion, turbine distress, ruptured bleed valve.

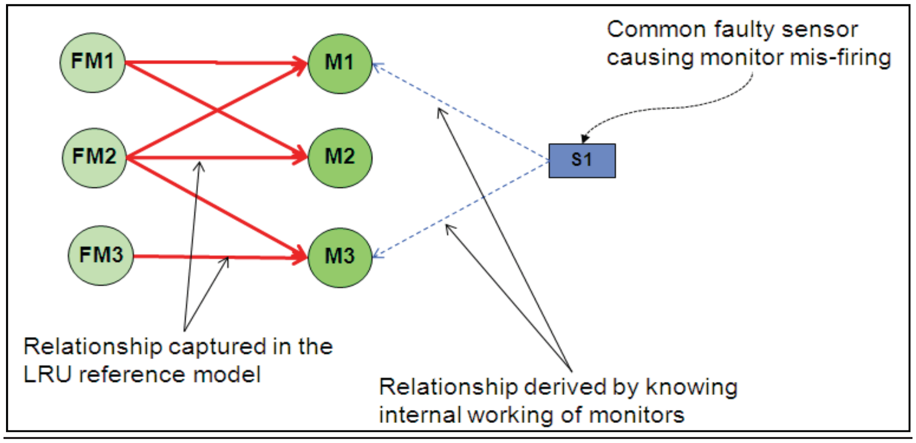

FM1
FM2
FM3
M1
M2
M3
S1
Common faulty sensor causing monitor mis-firing
Relationship captured in the LRU reference model
Relationship derived by knowing internal working of monitors

Figure 11. VIPR ConOps for faulty sensor induced intermittent event

VIPR does support multiple simultaneous faults. However, meta-rules within VIPR calculate the likelihood of such events. In this scenario, these failure modes are possible, but very unlikely. VIPR hypothesizes potential sensor fault. Using the reference model, VIPR identifies a set of sensors that is common to primary monitors associated with each fault condition. Through active query, it compares the engine installed temperature (T2) sensor with the aircraft installed temperature sensor and isolates the problem to an in-range bias of the engine T2 sensor.

4. Software prognostics triggered by a relatively benign fault. This benign fault, in this scenario, leads to a much more serious fault due to a software design flaw as reported by the Australian Transport Safety Bureau of a Boeing 777 on August 1, 2005 incident. VIPR keeps track of current contingency state (backup sensor, active I/O channel) to collect data surrounding these relatively routine events. The archived evidence can be used to rerun a portion of a V&V model with appropriate boundary conditions, and calculate likelihood of system-level failures if and when the backup sensor also fails. We are viewing this scenario in the context of prognostics and prevention schema, so that VIPR helps ensure that the much more serious event which happened in real life never comes to pass.

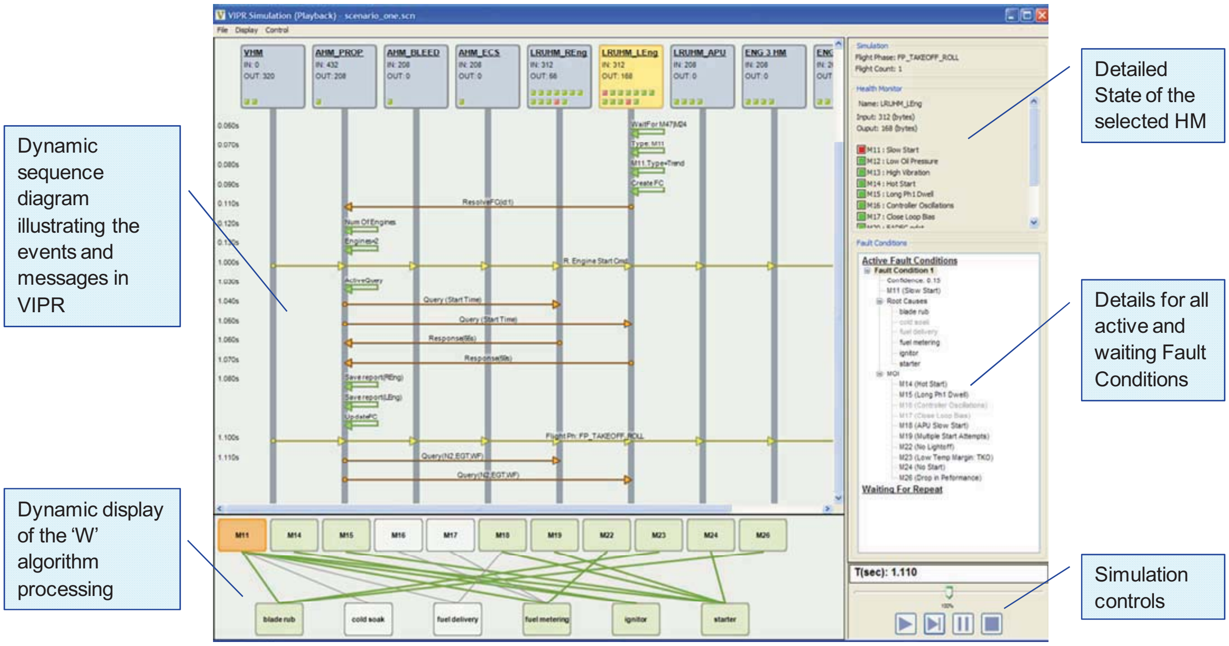
6.7 VIPR Demonstration

The demonstration of VIPR in year one illustrates VIPR's "plumbing". No reasoners are included in the first year, although a scripted illustration of how these reasoners will work is included. The demo screen is shown below in Figure 12.

The demo is what we call an animated Concept of Operations or ConOps. Using the simulator and the scenario scripts, the demonstration allowed us to "see" the VIPR design, highlight salient features, discover limitations and make course corrections. We used four scenarios to visualize the diagnostic and prognostic steps, message passing for active data query and hence illustrate how VIPR concepts work together to achieve the overall NASA IVHM goals.

Visualization of the VIPR architecture and internal working were done using (Refer to Figure 12):

1. A dynamic sequence diagram (upper left) that visualized the message passing within VIPR tiers following the information protocols. The senders and receivers are shown at the top of the window.
2. The window in the lower left illustrates the progression of the diagnostic computation using the "W-algorithm" (see [Biswas 6/2010] section 5.2.1).
3. The left middle window shows the details for all active fault conditions that describe the prevailing fault hypothesis.
4. The lower right is the window containing the controls for the demonstration.



Simulation
Flight Phase: FP_TAKEOFF_ROLL
Flight Count: 1

Health Monitor
Name: LRUHM_LEng
Input: 312 (bytes)
Ouput: 168 (bytes)

🟥 M11 : Slow Start
🟩 M12 : Low Oil Pressure
🟩 M13 : High Vibration
🟩 M14 : Hot Start
🟩 M15 : Long Ph1 Dwell
🟩 M16 : Controller Oscillations
🟩 M17 : Close Loop Bias

Fault Conditions

Active Fault Conditions
- Fault Condition 1
  - Confidence: 0.15
  - M11 (Slow Start)
  - Root Causes
    - blade rub
    - cold soak
    - fuel delivery
    - fuel metering
    - ignitor
    - starter
  - MOI
    - M14 (Hot Start)
    - M15 (Long Ph1 Dwell)
    - M16 (Controller Oscillations)
    - M17 (Close Loop Bias)
    - M18 (APU Slow Start)
    - M19 (Multiple Start Attempts)
    - M22 (No Lightoff)
    - M23 (Low Temp Margin: TKO)
    - M24 (No Start)
    - M26 (Drop in Performance)

Waiting For Repeat

T(sec): 1.110

Dynamic sequence diagram illustrating the events and messages in VIPR

Dynamic display of the 'W' algorithm processing

Detailed State of the selected HM

Details for all active and waiting Fault Conditions

Simulation controls

Figure 12. VIPR Demonstration

Other windows can be displayed during the progress of the demo. One of these is the EICAS display where any crew messages from VIPR can be displayed. Another is a multi-flight confidence plot associated with each fault condition. Finally, in support of next year's metrics evaluation, a window displaying statistics related to computational and networking resource usage for the scenario can be selectively displayed. Currently, this window displays the number of bytes transmitted, average message delay, message distribution by type, etc. for each of the VIPR's Health Manager nodes.

6.7.1 VIPR Demonstration Internals

Following the Honeywell 7-Step SMART process (see Figure 3) for the concept definition phase, the VIPR simulation focuses on concept of operations (CONOPS) simulation. The VIPR CONOPS simulation uses predefined scenarios to illustrate the expected operation of the VIPR system and is a Python application designed using a Model-View-Controller approach. The Model represents the current state of the VIPR system including all messages, monitors, and the processing state of each health manager.

The Controller provides a linkage between the model and the view and separates the model data from the visual representation. As the model state changes due to message traffic or monitors, the controller directs the view to update its presentation of the model. Likewise, the controller reflects user interactions, such as changing the state of a monitor, back to the model.

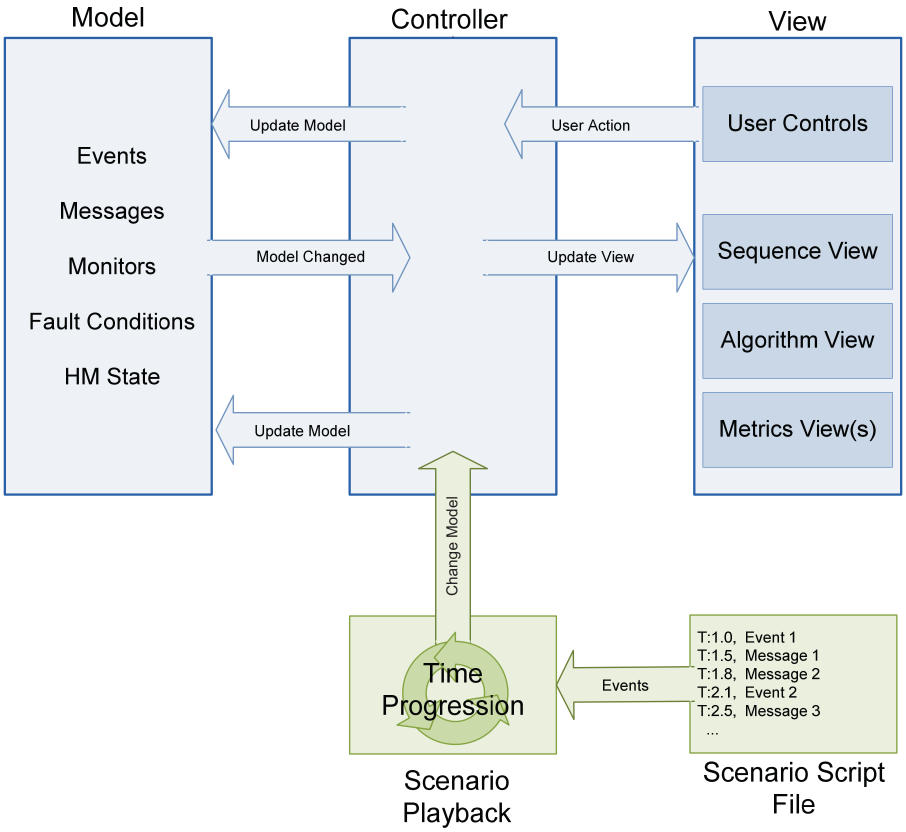


Figure 13. CONOPS Simulation Architecture

The View provides a visual representation of the model to the user. The CONOPS simulation divides the view into several display panels, each presenting a different perspective of the model (see Figure 12). The two primary views are the sequence view and the algorithm view. The sequence view illustrates the time sequenced message traffic and events as a dynamic sequence diagram, while the algorithm view shows the dynamic state of the 'W' algorithm for a health manager. Other, secondary views include performance metrics, pilot alert panel, and a fault condition plot.

Simulation execution is driven by a scenario script. The scenario contains a timeline of messages, fault conditions, monitor events, and process events representing the activity that is expected in a VIPR system. During playback of the scenario, events are applied to the model in time sequence according to the simulation clock time. The user can control the simulation playback speed or step through the simulation events using the simulation controller.

The next simulation phase will evolve the CONOPS simulation into an architectural model by replacing the scripted scenario events with a set of simulated health managers and monitors. The models will communicate via the VIPR Information Protocol message format over a simulated ARINC 624 bus.

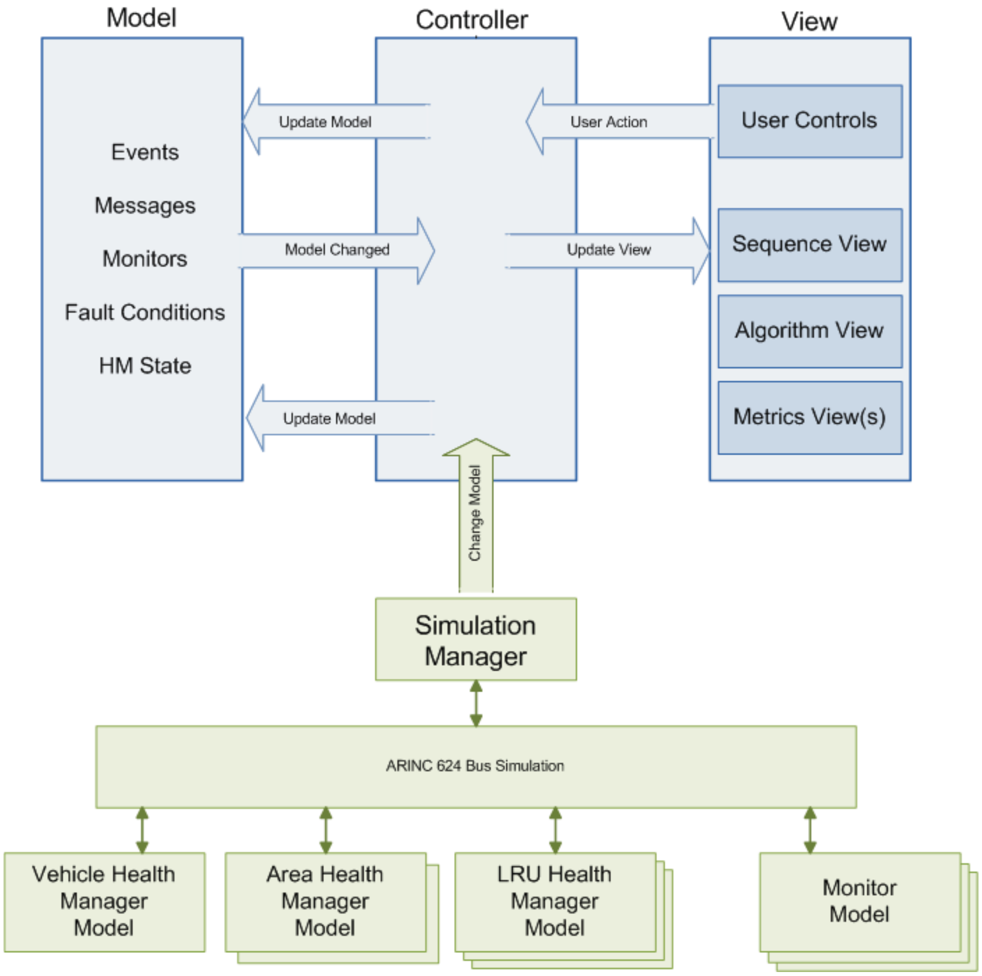


Figure 14. Architecture Simulation

The architectural model provides the opportunity to evaluate performance metrics such as data throughput and processor loading, along with validation of the system communication architecture and message formats.

6.8 Metrics for Benchmarking VIPR

A number of diagnostic and prognostic metrics exist, but these standards are defined for well-circumscribed algorithms that apply to small subsystems. For layered reasoners, such as VIPR, the overall performance cannot be evaluated by metrics solely directed toward timely detection and accuracy of estimation of the faults in individual components. Among other factors, the overall vehicle reasoner performance is governed by the effectiveness of the communication schemes between the different monitors and hierarchical reasoners in the architecture, and the ability to propagate and fuse relevant information to make accurate, consistent, and timely predictions at different levels of the reasoner hierarchy. An added functionality of this architecture is the ability of the vehicle- and area-level reasoners to generate specific queries for the component monitors. To address these issues, we have developed an extended set of diagnostic and prognostics metrics that can be used to evaluate the performance of the layered architecture. The metrics are summarized in the following tables.

Table 5. Detection and Diagnosis Metrics

Diagnostic coverage	• Identify test scenarios with faults that could not be detected and/or isolated with existing approaches and demonstrate VIPR's effectiveness for these scenarios
Accuracy	• Detection: false positive rate • Detection: false negative rate • Isolation: misclassification rate
Latency	• Time to detect • Time to isolate
Sensitivity	• Evaluate the metrics above in the presence of system uncertainty

Table 6. Prognosis Metrics

Prognostic coverage	• Identify test scenarios with faults that could not be predicted with existing approaches and demonstrate VIPR's effectiveness for these scenarios
Accuracy	• Error = predicted RUL – actual RUL • Average bias • Timeliness
Precision	• Estimate the size of the confidence interval associated with the RUL prediction
Sensitivity	• Evaluate the metrics above in the presence of system uncertainty

Table 7. Computational Metrics

Offline complexity analysis	• Worst- or average-case estimates of running time, memory, and communication bandwidth as a function of the size of the input • Number of software components • Number of links between software components • Number of inputs and outputs communicated • Size of code

Online profiling	• CPU execution times • CPU utilization • Network delays • Bandwidth Utilization • Amount of memory

A version of this report deliverable was also accepted at the PHM Conference to be held in Portland, OR in October 2010.

7 Recommendations for Future Work

We recommend that the next steps in the VIPR program include two parallel activities: Data Mining and Reasoner Coding. The Data Mining task includes the discovery of new relationships between symptoms and faults as well as more refined values of the parameters governing these relationships. The coding task will include detailed design and software implementation of the VIPR system reasoners for prognostics, diagnostics, and fusion.

Overall, there is a clear sequence for constructing the full aircraft reference model. Given the extensive understanding of propulsion and bleed subsystem health management, we recommend constructing their reference models first. The construction of reference models for other subsystems (software, actuators, and avionics) needs to be preceded by a data mining task.

The next steps would include integration of the reference models and the reasoner code within a simulation environment such as SMARTlab, demonstration of VIPR capabilities through a set of scenarios (section 6.6), collection of metrics (section 6.8), and trade space documentation.

Following a successful software demonstration, we recommend that select hardware be incorporated into the simulation environment to demonstrate VIPR's health management on real-world equipment.

8 Published Documents

The following documents were published under this contract:

Tim Felke, George D. Hadden, Dave Miller, Dinkar A. Mylaraswamy "Architectures for Integrated Vehicle Health Management", Proceedings of the AIAA Infotech@Aerospace 2010 Conference, April 2010, Atlanta, Georgia.

Xenofon Koutsoukos, Gautam Biswas, Dinkar A. Mylaraswamy, George D. Hadden, Daniel Mack, Doug Hamilton, "Benchmarking the Vehicle Integrated Prognostic Reasoner", PHM 2010, Portland, Oregon (to appear poster session).

Gautam Biswas, Xenofon Koutsoukos, Daniel Mack, Dinkar A. Mylaraswamy and George D. Hadden, "Vehicle Integrated Prognostic Reasoner (VIPR): A Hierarchical Reasoner for Aircraft Systems", Proceedings of the 2011 IEEE Aerospace Conference (to appear).

9 Referenced Documents

The following documents were referenced in the research and analysis conducted for this contract:

S. Abdelwahed and G. Karsai, “Notions of Diagnosability for Timed Failure Propagation Graphs”, IEEE Systems Readiness Technology Conference, pp. 643-648, 2006.

C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X.D. Koutsoukos, “Local Causal and Markov Blanket Induction Algorithms for Causal Discovery and Feature Selection for Classification, Part I: Algorithms and Empirical Evaluation”, Journal of Machine Learning Research 11 (Jan), pp. 171-234, 2010.

C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X.D. Koutsoukos. “Local Causal and Markov Blanket Induction Algorithms for Causal Discovery and Feature Selection for Classification, Part II: Analysis and Extensions”, Journal of Machine Learning Research. 11 (Jan), pp. 235−284, 2010.

D. Badenius, “Failure Rate/MTBF”, IEEE Transactions on Reliability, vol. R-19 (2), pp. 66-67, 1970.

R. Bharadwaj, C. Kulkarni, G. Biswas, and K. Kim, “Model-Based Avionics Systems Fault Simulation and Detection”, Proceedings of the AIAA Infotech@Aerospace 2010 Conference, April 2010, Atlanta, Georgia.

G. Biswas and T.S. Anand, “An Expert System Shell for Mixed Initiative Reasoning”, Journal of the Indian Institute of Science, Bangalore, India, 67, pp. 465-490, 1987.

G. Biswas, R. Kapadia, and X.W. Yu, “Combined Qualitative–Quantitative Steady-State Diagnosis of Continuous-Valued Systems”, IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, vol. 27, no. 2, pp. 167-185, March 1997.

G. Biswas, X. Koutsoukos, D. Mylaraswamy, G.D. Hadden “Tools and Technology Concept of Operations”, Technical Report, submitted by Honeywell to NASA (CDRL Sequence Number: 4.1.07), June 2010.

G. Biswas and T.S. Anand, “Using the Dempster-Shafer Scheme in a Mixed-Initiative Expert System Shell”, Uncertainty in Artificial Intelligence 3, L.N. Kanal, T.S. Levitt, and J.F. Lemmer, eds., Elsevier Science Publishers (North Holland), Amsterdam, pp. 223-239, 1989.

R.M. Button and A. Chicatelli, “Electrical Power System Health Management”, Proc. 1st International Forum on Integrated System Health Engineering and Management in Aerospace, Napa, CA, November 2005.

C.S. Byington, M. Watson, P. Kalgren, and R. Safa-Bakhsh, “Metrics Evaluation and Tool Development for Health and Usage Monitoring System Technology”, Third International Conference on Health and Usage Monitoring - HUMS2003, Melbourne, Australia, G.F Forsyth (editor), pp. 27-35, 2003.

C.S. Chao, D.L. Yang, and A.C. Liu, “An Automated Fault Diagnosis System Using Hierarchical Reasoning and Alarm Correlation”, Journal of Network and Systems Management, 9(2), pp. 183-202, 2001.

L. Chittaro and R. Ranon, “Hierarchical Model-Based Diagnosis Based On Structural Abstraction”, Artificial Intelligence, 155, pp. 147-182, 2004.

I. Cohen, M. Goldszmidt, T. Kelly, J. Symons, and J.S. Chase, “Correlating Instrumentation Data to System States: A Building Block for Automated Diagnosis and Control”, Sixth Symposium on Operating Systems Design and Implementation, San Francisco, CA, Dec. 6-8.Chen, K. (2006) TAN Trees for fMRI Brain Image Classification, Carnegie Mellon University, 2004.

J. Colclough, “The PS-MRS Reference Model”, U.S. Army Contract W56HZV-05-C-0724.

M. Daigle, X. Koutsoukos, and G. Biswas, “Distributed Diagnosis in Formations of Mobile Robots”, IEEE Transactions on Robotics, vol. 23, no. 2, pp. 353-369, 2007.

R. Davis and W. Hamscher, “Model-Based Reasoning: Troubleshooting”, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1988.

J. de Kleer and B.C. Williams, “Diagnosing Multiple Faults”, Artificial Intelligence, vol. 32, pp. 97-130, 1987.

R. Debouk, S. Lafortune, and D. Teneketzis, “Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems”, Discrete Event Dynamic System: Theory and Applications, 10(1/2), pp. 33–86, January 2000.

T. Felke, “Application of Model-Based Diagnostics Technology on the Boeing 777 Airplane”, IEEE Aero Conf., Big Sky, MT. 1994.

T. Felke, “Knowledge Management for Maintenance of Complex Systems”, https://dashlink.arc.nasa.gov/topic/integrated-diagnostics, Briefing to NASA ARC, April 21, 2009.

P. Frank, “Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy - A Survey and Some New Results”, Automatica, vol. 26, no. 3, pp. 459-474, 1990.

N. Freidman, D. Geiger, and M. Goldszmidt (1997), "Bayesian Network Classifiers”, Machine Learning, 29: 131-163.

J. J. Gertler, “Fault Detection and Diagnosis in Engineering Systems”, New York, NY, Marcel Dekker, Inc., 1998.

J. Goericke, K. Moeckly, and A. Stramiello, “An Embeddable, Physics-Based Approach for Continuous Performance Deterioration Detection and Power Assurance for Turboshaft Engines”, AHS 65th Annual Forum, Grapevine, TX, 2009.

J. Gordon and E.H. Shortliffe, “The Dempster-Shafer Theory of Evidence”, Chapter 13, Rule-Based Expert Systems: The MYCIN Experiments of the Heuristic Programming Project, Addison-Wesley, Reading, MA, 1984.

V. Guralnik, D. Mylaraswamy, and H. Voges, “On Handling Dependent Evidence and Multiple Faults in Knowledge Fusion for Engine Health Management”, 2006 IEEE Aerospace Conference, 2006.

V. Guralnik and D. Mylaraswamy, “A Stochastic Theory for Evidence Aggregation”, First PHM Conference, October 2008.

G. Hadden, R. Magnuson, B.J. Barton, G. Shao, “A Method to Perform Prognostics Through Prediction of Diagnostic Indicators”, US Patent Application, 2009.

ISO 13374 Standard http://www.iso.org/iso/catalogue_detail?csnumber=21832

B. Jambor (Program Manager, Lockheed Martin Space Systems Company), “Integrated System Vehicle Health Management Architecture Design”, AFRL Contract FA8650-05-5503-0001.

X. Koutsoukos, G. Biswas, D. Mylaraswamy, G. Hadden, D. Mack, D. Hamilton, “Benchmarking the Vehicle Integrated Prognostic Reasoner (VIPR)”, Technical Report, submitted by Honeywell to NASA (CDRL Sequence Number: 4.1.06), April 2010.

T. Kurtoglu, O. Mengshoel, and S. Pol, “A Framework for Systematic Benchmarking of Monitoring and Diagnostic Systems”, International Conference on Prognostics and Health Management Conference, Denver, CO, 2008.

P.J. Mosterman and G. Biswas, “Diagnosis of Continuous Valued Systems in Transient Operating Regions”, IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 29, no. 6, pp. 554-565, 1999.

D. Mylaraswamy and G. Hadden, “Architecture Recommendations: NASA VIPR Program”, Technical Report, submitted by Honeywell to NASA (CDRL Sequence Number: 4.1.02), December 2009.

D. Mylaraswamy, D. Hamilton, and G. Hadden, “Concept of Operations: NASA VIPR Program”, Technical Report, submitted by Honeywell to NASA (CDRL Sequence Number: 4.1.04), January 2010.

D. Mylaraswamy, L. Olson, and E. Nwadiogbu, “Engine Performance Trending: Practical Insights”, Trans. Of the Institute of Measurement and Control, 31, pp. 341-354, 2009.

D. Mylaraswamy, P. Bullemer, and K. Emigholz, “Fielding a Multiple State Estimator Platform”, National Petroleum and Refiners Association Computing Conference, October 2000.

D. Mylaraswamy, D. Hamilton, and G. Hadden, “Information Protocols: NASA VIPR Program”, Technical Report, submitted by Honeywell to NASA (CDRL Sequence Number: 4.1.03), December 2009.

J. Pearl, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference”, Morgan Kaufmann, San Mateo, CA, 1988.

R. Reiter, “A Theory of Diagnosis from First Principles”, Readings in Model-based Diagnosis, pp. 29-48, 1992.

A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher, “Metrics for Evaluating Performance of Prognostic Techniques”, in International Conference on Prognostics and Health Management (PHM08), Denver CO, 2008.

A. Saxena, J. Celaya, B. Saha, and K. Goebel, “Prognostics Performance Evaluation”, NASA 2009 Aviation Safety Conference.

G. Schwarz, “Estimating the Dimension of a Model”, The Annals of Statistics, 6 (2), pp. 461-464, 1978.

G. Shafer, “A Mathematical Theory of Evidence”, Princeton Univ. Press, 1976.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 01- 05 - 2011	**2. REPORT TYPE** Contractor Report	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE

Vehicle Integrated Prognostic Reasoner (VIPR) 2010 Annual Final Report

5a. CONTRACT NUMBER

NNL09AD44T

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hadden, George D.; Dinkar, Mylaraswamy; Schimmel, Craig; Biswas, Gautam; Koutsoukos, Xenofon; Mack, Daniel

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

645846.02.07.07

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/CR-2011-217147

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 06
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES

Langley Technical Monitor: Eric G. Cooper

14. ABSTRACT

Honeywell's Central Maintenance Computer Function (CMCF) and Aircraft Condition Monitoring Function (ACMF) represent the state-of-the art in integrated vehicle health management (IVHM). Underlying these technologies is a fault propagation modeling system that provides nose-to-tail coverage and root cause diagnostics. The Vehicle Integrated Prognostic Reasoner (VIPR) extends this technology to interpret evidence generated by advanced diagnostic and prognostic monitors provided by component suppliers to detect, isolate, and predict adverse events that affect flight safety. This report describes year one work that included defining the architecture and communication protocols and establishing the user requirements for such a system. Based on these and a set of ConOps scenarios, we designed and implemented a demonstration of communication pathways and associated three-tiered health management architecture. A series of scripted scenarios showed how VIPR would detect adverse events before they escalate as safety incidents through a combination of advanced reasoning and additional aircraft data collected from an aircraft condition monitoring system. Demonstrating VIPR capability for cases recorded in the ASIAS database and cross linking them with historical aircraft data is planned for year two.

15. SUBJECT TERMS

Advanced Diagnostics, Condition Monitoring, Prognostics, Vehicle Health Management, Vehicle-Level Reasoning

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER ***(Include area code)***
U	U	U	UU	35	(443) 757-5802