PHM Enabled Autonomous Propellant Loading Operations

Mark Walker
D2K Technologies
5062 Nighthawk Way
Oceanside, CA 92056
760-685-4028
Mark.Walker@d2ktech.com

Fernando Figueroa
NASA Stennis Space Center
Engineering and Test Directorate
Stennis Space Center, MS 39529
228-688-2482
fernando.figueroa@nasa.gov

Jaime Toro-Medina
NASA Kennedy Space Center
Gas and Fluid Systems
Kennedy Space Center, FL 39529
321-867-6445
jaime.a.toromedina@nasa.gov

Abstract— **The utility of Prognostics and Health Management (PHM) software capability applied to Autonomous Operations (AO) remains an active research area within aerospace applications. The ability to gain insight into which assets and subsystems are functioning properly, along with the derivation of confident predictions concerning future ability, reliability, and availability, are important enablers for making sound mission planning decisions. When coupled with software that fully supports mission planning and execution, an integrated solution can be developed that leverages state assessment and estimation for the purposes of delivering autonomous operations. The authors have been applying this integrated, model-based approach to the autonomous loading of cryogenic spacecraft propellants at Kennedy Space Center.**

TABLE OF CONTENTS

1. INTRODUCTION

The demand for autonomous management and control of aerospace systems for both ground and flight operations is increasing at a fervent pace. This includes systems that are required to perform according to mission objectives when there are no humans available for making decisions. The demand is also driven by manned systems which can be operated more safely, more cost-effectively, or more conveniently when operated without direct human intervention. There are a number of good examples of emerging applications of both types of autonomous systems within the Department of Defense [12]-[15], and [17]. Other examples are prevalent within NASA, including applications for autonomous vehicles and unmanned spacecraft systems [1] and [3], as well as manned flight operations involving systems for which ground based command and control is impossible and/or for which control by astronauts is deemed undesirable [4]. This paper describes ongoing NASA research on the application of autonomous operations for use in ground-based spacecraft propellant loading systems for which safety is a major concern but for which the elimination of direct human involvement is desired.

Because AO systems are responsible for acting on decisions based on what is happening externally (external awareness), as well as what is happening internally (internal awareness), the artificial intelligence technologies associated with system level Prognostics and Health Management have emerged as key enabling technologies for delivering autonomous operations. It is important to note, however, that in most cases the delivery of such systems has been plagued due to excessive cost, inability to deliver, and concerns about safety or their lack of determinism. The challenges associated with certification of safety critical systems and the difficulties regarding software requirements verification and validation (V&V) are well documented [3], [14], and [17]-[18] and continue to be a major impediment to the deployment of autonomous systems.

This paper presents a novel architecture that incorporates a generic, re-usable model-based infrastructure for building AO systems that readily leverage PHM capability. The approach and architecture also support a reliability-centered design and implementation that helps cut cost, shorten development time, expedite SQA testing, and facilitate requirements V&V. The described work is associated with applying this platform and methodology for delivering the objectives associated with autonomous spacecraft propellant loading (APL).

Section 2 provides an overview of PHM and its linkage to AO, while also providing some background on the design and architecture of our generic, model-based reasoning platform. Section 3 is dedicated to describing the application of this technology to APL at several test facilities at Kennedy Space Center (2011 – present). Section 4 provides a summary of failure modes tested during the verification and validation process of the AO system for APL, and Section 5 provides concluding remarks and suggestions for future research.

2. PHM ENABLED AUTONOMY

Historically, implementation of autonomous software systems has taken many forms. Most typically, these have been hard-coded, developed from scratch, one-off solutions that manage and control autonomy through policy-based logic applied across whatever sensor data is available. Lately,

978-1-5090-1613-6/17/$31.00 ©2017 IEEE

more sophisticated solutions have emerged involving dynamic modeling and prediction, as well as the use of empirical learning and statistical pattern recognition algorithms [2], [8], [13], and [20]-[22]. However, AO systems attempting to leverage these technologies face challenges in identifying which of these technologies should be used, how they should be used, and when they should be used, based on varying mission circumstances [17]-[18]. This under-addressed aspect of designing AO software has proven to be a bigger issue than anticipated by researchers, creating confusion and delivering limited performance. This is further exacerbated by the fact that the problems aren't fully embraced or even understood until late in the design cycle [16]. For AO system developers, the choice for which technologies and approaches to utilize is daunting, as is depicted in Figure 1.

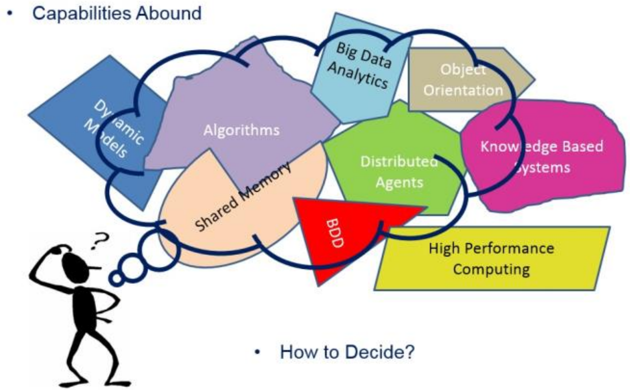


Figure 1. Choosing the right AO technologies

One specific issue with the historical approaches to AO applications is that they are very often hard-wired to recognize and respond to specific anomalous behaviors tied to specific components within the controlled system. Ideally, an AO system would obtain good information about the capacity of the controlled system to perform its required functions – expressed in terms of the system, the functional requirements of its components, and the impact to the system when the components fail. In other words, what would be beneficial is the leveraging of an object-oriented PHM platform equipped with a modeling infrastructure that supports a reliability focused design [7] and [14] of the target AO system.

PHM as an Enabler for Autonomy

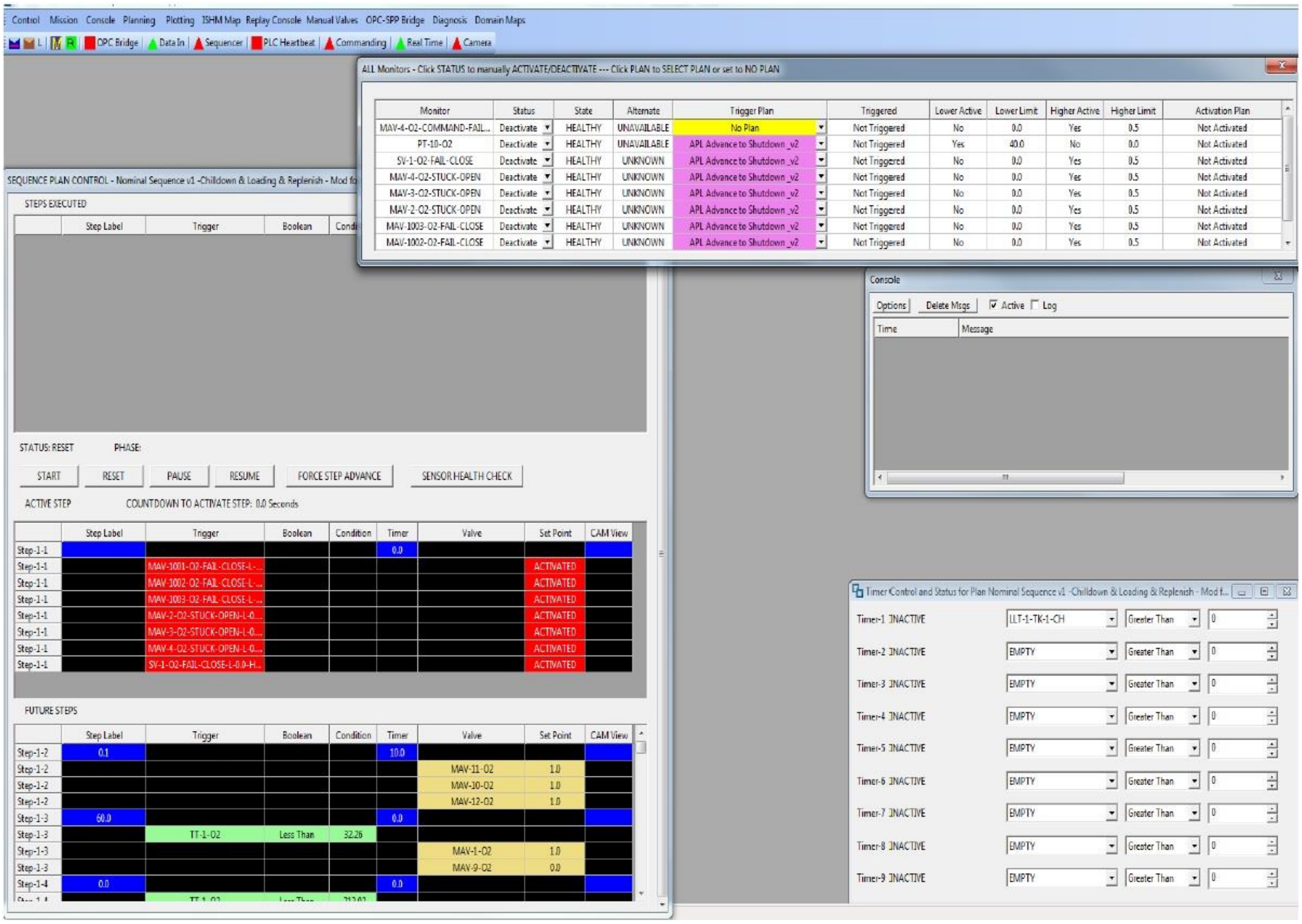
Among the emerging technologies that promise to be key enablers for delivering AO are those associated with systems that provide Prognostics and Health Management (PHM) [2], [8], and [14]. PHM systems are dedicated to assessing or estimating both present and future condition (and availability) of mission critical systems and their components. These assessments are usually made based on available sensor data, along with (in the case of advanced systems) the operating context of the system according to mission objectives [14]. In some cases [15] and [20]-[22], platforms are used that support reasoning over an object-oriented model of the system built from libraries of generic equipment classes. If the modeling language is sufficiently flexible, it is possible to develop rich context-based diagnostic software that delivers strong and reliable situation awareness [19] to end-users. Enhanced situation awareness to end-users and operators is made even more valuable when combined with automated, real-time decision support based on human expertise. In some cases, the PHM system may be responsible for delivering automated condition-based maintenance (CBM) support through expedited diagnosis, automated workflow for troubleshooting, and integration with supply chain and depot inventory systems [2], [3], and [17]. These benefits help stakeholders and maintainers maximize availability, reduce downtime, and lower the logistical costs of system operation.

When coupled with software that supports step-wise mission plan specification and execution, a PHM system can readily be integrated into an AO architecture for supporting the internal awareness requirements of autonomous operations. The authors have been addressing these requirements as part of their work in delivering a PHM enabled AO system for cryogenic propellant loading of spacecraft flight tanks (refer to Section 3) [20]-[22]. An example dialogue from that system showing the user-specified step-wise mission plan for the multi-phase loading operation of cryogenic propellants is shown in Figure 2. In this example, autonomous control of valve commands are being invoked based on mission specific decision criteria that are evaluated based on modeled system state, sensor data, and operational context. If certain anomalies are detected, or if context specific redline conditions are satisfied, an automated system safing plan is invoked without requiring human intervention.

Obstacles to Achieving AO

In many cases the delivery of such AO systems has been derailed due to concerns about safety or their lack of determinism. The challenges associated with certification of autonomous operations for safety critical systems and the difficulties regarding software requirements verification and validation are well documented [3] and continue to be a major impediment to the deployment of autonomous systems. There continues to be widespread mistrust of autonomous software [18] and, similar to mistrust of driverless vehicles, only demonstrated real-world performance of AO systems will likely overcome the pessimism.

Other obstacles experienced by those attempting to deploy sophisticated AO solutions are related to the complexity of delivering and maintaining AO solutions on-time and within budget. The specific challenges and exercised approaches at



ALL Monitors - Click STATUS to manually ACTIVATE/DEACTIVATE --- Click PLAN to SELECT PLAN or set to NO PLAN

Monitor	Status	State	Alternate	Trigger Plan	Triggered	Lower Active	Lower Limit	Higher Active	Higher Limit	Activation Plan
MAV-4-O2-COMMAND-FAIL...	Deactivate	HEALTHY	UNAVAILABLE	No Plan	Not Triggered	No	0.0	Yes	0.5	Not Activated
PT-10-O2	Deactivate	HEALTHY	UNAVAILABLE	APL Advance to Shutdown _v2	Not Triggered	Yes	40.0	No	0.0	Not Activated
SV-1-O2-FAIL-CLOSE	Deactivate	HEALTHY	UNKNOWN	APL Advance to Shutdown _v2	Not Triggered	No	0.0	Yes	0.5	Not Activated
MAV-4-O2-STUCK-OPEN	Deactivate	HEALTHY	UNKNOWN	APL Advance to Shutdown _v2	Not Triggered	No	0.0	Yes	0.5	Not Activated
MAV-3-O2-STUCK-OPEN	Deactivate	HEALTHY	UNKNOWN	APL Advance to Shutdown _v2	Not Triggered	No	0.0	Yes	0.5	Not Activated
MAV-2-O2-STUCK-OPEN	Deactivate	HEALTHY	UNKNOWN	APL Advance to Shutdown _v2	Not Triggered	No	0.0	Yes	0.5	Not Activated
MAV-1003-O2-FAIL-CLOSE	Deactivate	HEALTHY	UNKNOWN	APL Advance to Shutdown _v2	Not Triggered	No	0.0	Yes	0.5	Not Activated
MAV-1002-O2-FAIL-CLOSE	Deactivate	HEALTHY	UNKNOWN	APL Advance to Shutdown _v2	Not Triggered	No	0.0	Yes	0.5	Not Activated

SEQUENCE PLAN CONTROL - Nominal Sequence v1 -Chilldown & Loading & Replenish - Mod fo

STEPS EXECUTED

	Step Label	Trigger	Boolean	Condi

STATUS: RESET PHASE:

ACTIVE STEP COUNTDOWN TO ACTIVATE STEP: 0.0 Seconds

	Step Label	Trigger	Boolean	Condition	Timer	Valve	Set Point	CAM View
Step-1-1					0.0			
Step-1-1		MAV-1001-O2-FAIL-CLOSE-L-...					ACTIVATED	
Step-1-1		MAV-1002-O2-FAIL-CLOSE-L-...					ACTIVATED	
Step-1-1		MAV-1003-O2-FAIL-CLOSE-L-...					ACTIVATED	
Step-1-1		MAV-2-O2-STUCK-OPEN-L-0....					ACTIVATED	
Step-1-1		MAV-3-O2-STUCK-OPEN-L-0....					ACTIVATED	
Step-1-1		MAV-4-O2-STUCK-OPEN-L-0....					ACTIVATED	
Step-1-1		SV-1-O2-FAIL-CLOSE-L-0.0-H...					ACTIVATED	

FUTURE STEPS

	Step Label	Trigger	Boolean	Condition	Timer	Valve	Set Point	CAM View
Step-1-2	0.1				10.0			
Step-1-2						MAV-11-O2	1.0	
Step-1-2						MAV-10-O2	1.0	
Step-1-2						MAV-12-O2	1.0	
Step-1-3	60.0				0.0			
Step-1-3		TT-1-O2	Less Than	32.26				
Step-1-3						MAV-1-O2	1.0	
Step-1-3						MAV-9-O2	0.0	
Step-1-4	0.0				0.0			

Console

☑ Active ☐ Log

Time	Message

Timer Control and Status for Plan Nominal Sequence v1 -Chilldown & Loading & Replenish - Mod f...

Timer-1 INACTIVE	LLT-1-TK-1-CH	Greater Than	0
Timer-2 INACTIVE	EMPTY	Greater Than	0
Timer-3 INACTIVE	EMPTY	Greater Than	0
Timer-4 INACTIVE	EMPTY	Greater Than	0
Timer-5 INACTIVE	EMPTY	Greater Than	0
Timer-6 INACTIVE	EMPTY	Greater Than	0
Timer-7 INACTIVE	EMPTY	Greater Than	0
Timer-8 INACTIVE	EMPTY	Greater Than	0
Timer-9 INACTIVE	EMPTY	Greater Than	0

Figure 2. Mission Plan and Execution Dialogue

resolving them for AO of APL at KSC are discussed in more detail in Section 4.

Object-Oriented, Model-based PHM

In the past two decades, the boundaries of PHM systems have been heavily researched [11] and [12]-[16]. Across that time period there have been increasing numbers of successfully deployed PHM systems applied to a varying number of commercial and government applications. One of the differentiating characteristics of successfully deployed PHM systems is the leveraging of real-time system state models coupled with object-oriented, model-based procedural reasoning. This approach enables the developer to employ system health assessment based on class behaviors built around first principles rather than on hard-coded limits. When combined with modeling support for abstract object relationships, this methodology also lends itself to assessing system health based on mission objectives.

With object orientation, it is possible to architect the PHM system in a layered approach, where each layer exists at a higher level of abstraction. As depicted in Figure 3, an 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 position switch state changes that prompt an update to the 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 and is driven by Failure Modes and Effects Analysis (FMEA) [9]. 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

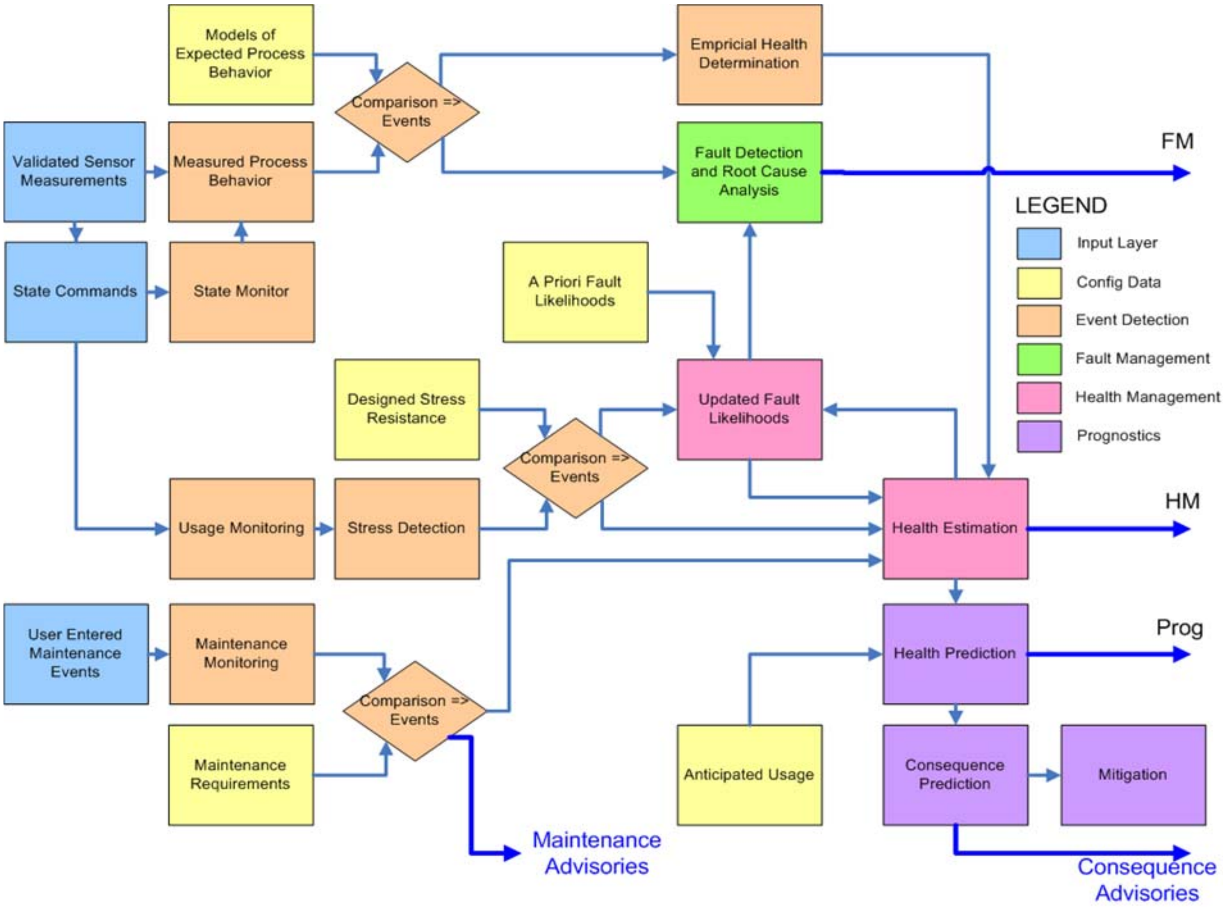


Figure 3. Layered PHM Architecture

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. The fault models are made available using causal directed graphing technology [10].

Model-based Reasoner as an Execution Platform

The aggregation of the expert system modeling objects, coupled with their object-oriented state assessment, functional capability assessment, event detection, anomaly detection, and remaining-useful-life (RUL) methods, all runs on top of the real-time execution engine in concert with the stepping of the mission plan. In this way, the assessment of system state and consideration of response actions using all available information becomes the heartbeat of the AO system. Working lock-step with mission execution, the model-based reasoner guides the real-time system through state space. The block diagram shown in Figure 4 depicts the role of the model-based reasoner as an execution platform.

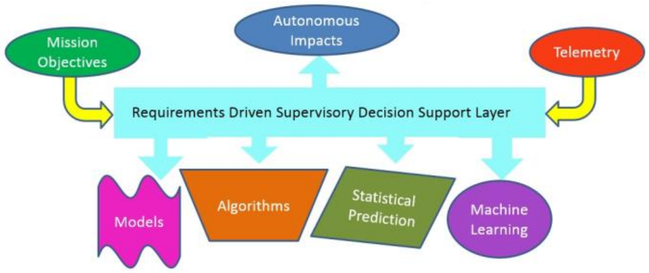


Figure 4. Model-based Execution Platform

Leveraging Object-Orientation for Requirements Assurance

Another benefit that can be leveraged from modern PHM systems is the incorporation of logic that evaluates the reliability-focused performance of the system based on comparisons to its required functions. Object-oriented system modeling capabilities equipped with representation of abstract object relationships lends itself well for this purpose. The power of this approach becomes even more evident when used to assist in software requirements verification and validation. The authors have employed these techniques for improving quality assurance through improved functional and requirements testing of the APL system.

3. Autonomous Propellant Loading

The loading of cryogenic propellants from storage tanks into spacecraft flight tanks immediately prior to launch is, while a recognizably dangerous activity, a well-known and understood operation with thousands of launch hours of experience acquired over the past five decades of launch operations at the various DoD and NASA launch facilities [5] and [6]. Relatively recent research [4] has been performed at the Cryogenics Test Laboratory (CTL, Kennedy Space Center) to increase the level of understanding of the physics involved in the transfer of cryogenic commodities. The CTL research also seeks improved understanding of the physics associated with any hazards associated with system failures, while gaining insight into the optimal control of the various remotely operated valves used to evoke efficient transfer. A photo of the Simulated Propellant Loading System (SPLS) at the CTL is shown in Figure 4.

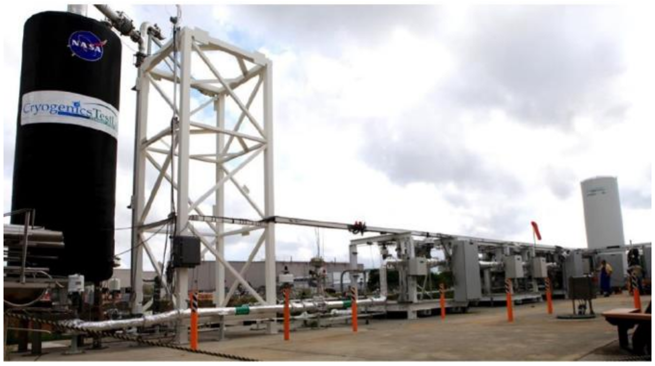


Figure 4. SPLS System at the KSC CTL

The authors were privileged to provide support for building a PHM enabled decision support system for the SPLS from 2011-2014. In order to maximize efficiency and safety of the autonomous transfer, vigilant monitoring is required of all hydraulic parameters, the position feedback of controlled valves, and a real-time comparison between measured and expected values. As a result of applying PHM techniques at SPLS, several potentially serious otherwise disguised issues with leaking isolation valves were revealed using correlation coefficients derived from delta pressure measurements. In addition, demonstration of PHM guided valve command sequencing was also performed by the authors at KSC's CTL [21].

Autonomous Operations System for APL

For future launch systems, it is desired to achieve even more autonomy over the cryogenic loading process, equipping the system with the ability for autonomous sequencing of valve commands and support for automated response to monitored red-line (abort) conditions. This has led to the creation of the Autonomous Operations System (AOS), a generic PHM enabled AO development platform being applied to APL at KSC. The intent of the APL AO system is to provide fully autonomous control and operation of the loading of cryogenic propellants between a Universal Propellant Service System (UPSS) and the Iron Rocket, a hardware emulation of a spacecraft's three stage flight tanks [20]. The AO system is required to sequence fill, vent, and purge valves throughout the various phases of propellant loading process (chilldown, slow and fast fill, and replenish), all while maintaining a close monitor on hydraulic conditions through the loading system and its various electronic and pneumatic control systems.

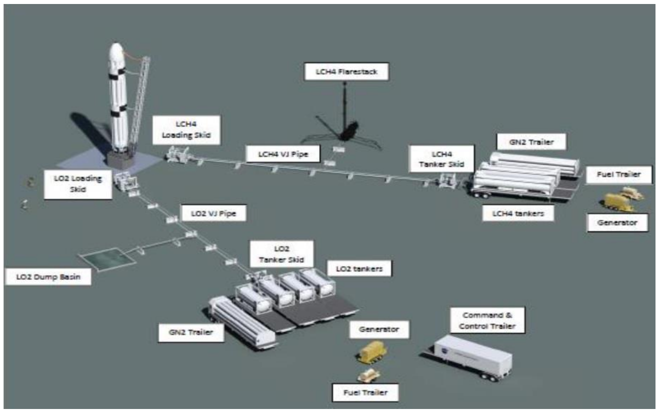


Figure 5. APL Conceptual Diagram

Detection and Distinction of Cryogenic Valve Anomalies

Among other things, the APL AO system is expected to use model-based approaches for detecting FMEA-derived valve state anomalies and distinguishing (without operator intervention) the difference between actual valve anomalies (stuck open, fails to seat, doesn't respond to commands) and valve position feedback sensor failures. This is especially important during certain operational contexts – for example, if a commanded fill valve experiences a fault during fast fill just as one of the flight tanks is approaching 90% full, the difference between a stuck valve and a "lying" position sensor becomes a safety critical distinction. In other cases, a valve which fails to respond to a command that is isolated by the model-based PHM system to a faulty position sensor during a non-time-sensitive operation may be appropriately ignored by the AO system, allowing for the loading process to continue without costly interruption, postponement, or outright abort.

AO for APL Integrated Software Architecture

The PHM enabled AO system used for developing the APL system was architected using Gensym's G2 expert system development platform and integrated with a gateway providing APL telemetry according to the Space Packet Protocol (SPP). The system was also integrated and tested against a physics based simulation system developed by an independent team and intended to be used for software V&V. Domain models for both the UPSS and DTS sides of the APL were built using previously developed generic modeling libraries and generic fault models were employed to detect and analyze failure modes as specified by FMEA.

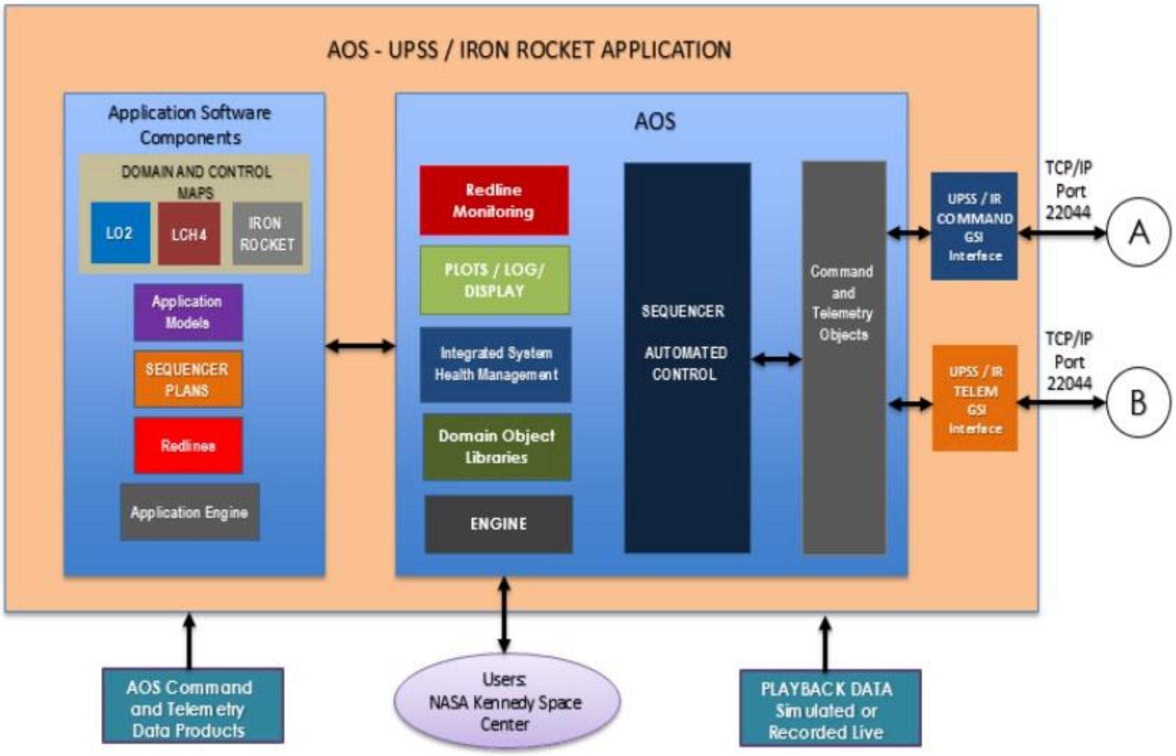

AOS - UPSS / IRON ROCKET APPLICATION
Application Software Components
DOMAIN AND CONTROL MAPS
LO2
LCH4
IRON ROCKET
Application Models
SEQUENCER PLANS
Redlines
Application Engine
AOS
Redline Monitoring
PLOTS / LOG/ DISPLAY
Integrated System Health Management
Domain Object Libraries
ENGINE
SEQUENCER
AUTOMATED CONTROL
Command and Telemetry Objects
UPSS / IR COMMAND GSI Interface
UPSS / IR TELEM GSI Interface
TCP/IP Port 22044
A
TCP/IP Port 22044
B
AOS Command and Telemetry Data Products
Users NASA Kennedy Space Center
PLAYBACK DATA Simulated or Recorded Live

Figure 6. AOS Application Architecture

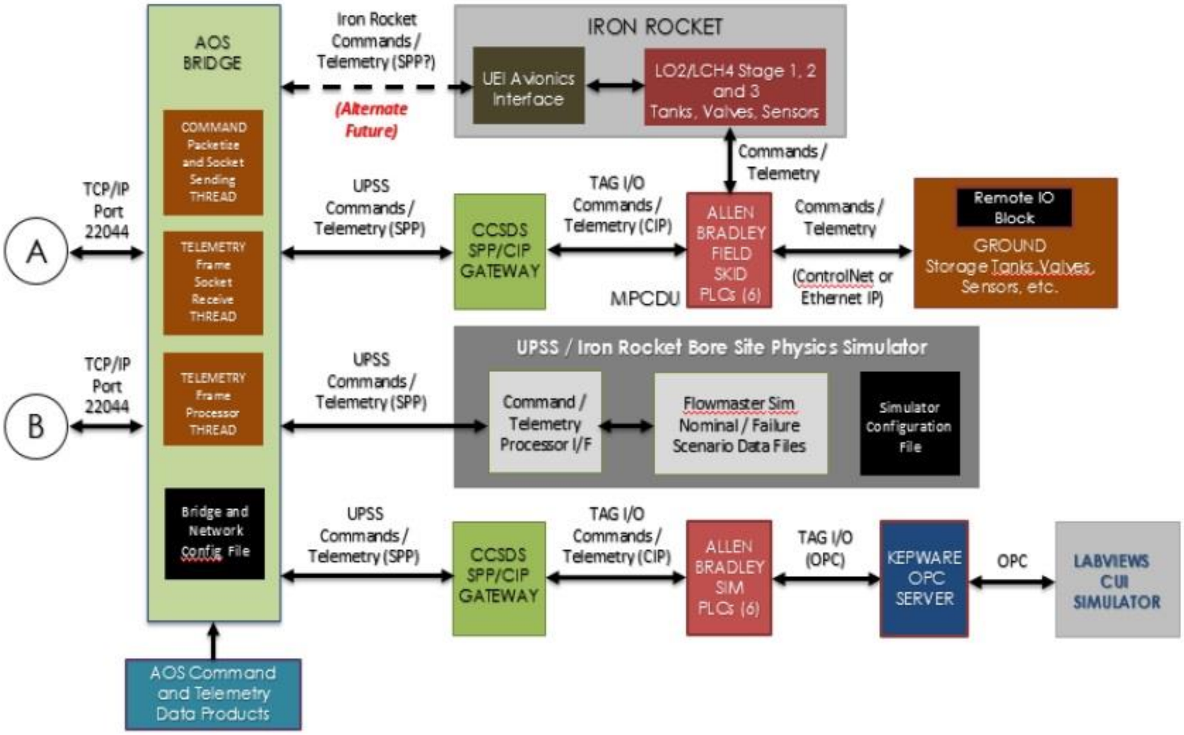

AOS BRIDGE
COMMAND Packetize and Socket Sending THREAD
TELEMETRY Frame Socket Receive THREAD
TELEMETRY Frame Processor THREAD
Bridge and Network Config File
TCP/IP Port 22044
A
TCP/IP Port 22044
B
Iron Rocket Commands / Telemetry (SPP?)
(Alternate Future)
IRON ROCKET
UEI Avionics Interface
LO2/LCH4 Stage 1, 2 and 3 Tanks, Valves, Sensors
Commands / Telemetry
UPSS Commands / Telemetry (SPP)
CCSDS SPP/CIP GATEWAY
TAG I/O Commands / Telemetry (CIP)
ALLEN BRADLEY FIELD SKID PLCs (6)
MPCDU
Commands / Telemetry
(ControlNet or Ethernet IP)
Remote IO Block
GROUND Storage Tanks,Valves, Sensors, etc.
UPSS Commands / Telemetry (SPP)
UPSS / Iron Rocket Bore Site Physics Simulator
Command / Telemetry Processor I/F
Flowmaster Sim Nominal / Failure Scenario Data Files
Simulator Configuration File
UPSS Commands / Telemetry (SPP)
CCSDS SPP/CIP GATEWAY
TAG I/O Commands / Telemetry (CIP)
ALLEN BRADLEY SIM PLCs (6)
TAG I/O (OPC)
KEPWARE OPC SERVER
OPC
LABVIEWS CUI SIMULATOR
AOS Command and Telemetry Data Products

Figure 7. DTS Application Architecture

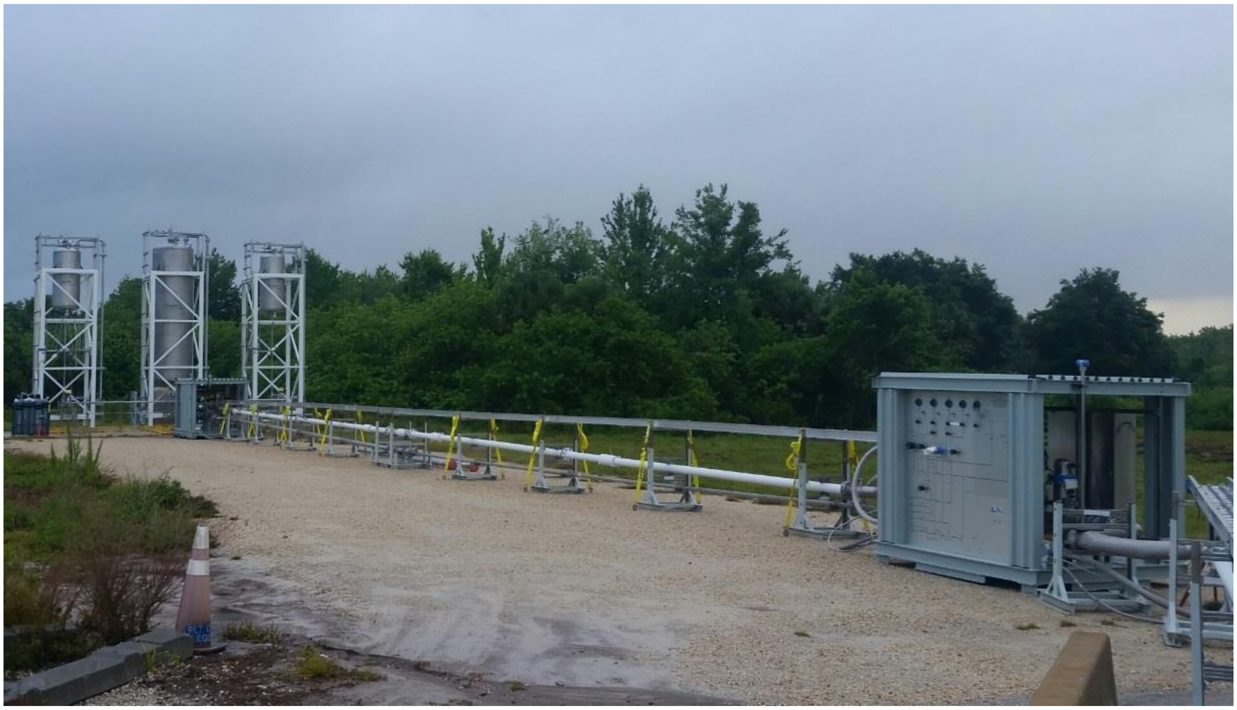


Figure 8. Valve Skids and Iron Rocket Tanks as part of the Autonomous Propellant Loading Test Site

Architecture diagrams for each half of the AO for APL are shown in Figures 6 and 7. The AOS consists of a generic part and an application (mission) specific part. Included in the generic part are the Domain Object Libraries, the generic equipment health assessment and fault libraries, Red Line Monitoring support, the Mission Planning Specification and Execution engine, generic Sequencer and autonomous control software, and generic command and telemetry objects [20]. Included in the DTS application specific part are the modeled APL system domains, the various APL control screens, the configured APL specific Mission Plans, and APL specific redline monitors. An image of the APL site (under construction) is depicted in Figure 8.

A screenshot of the combined UPSS and Iron Rocket domain models over which the APL SO system reasons is depicted in Figure 9.

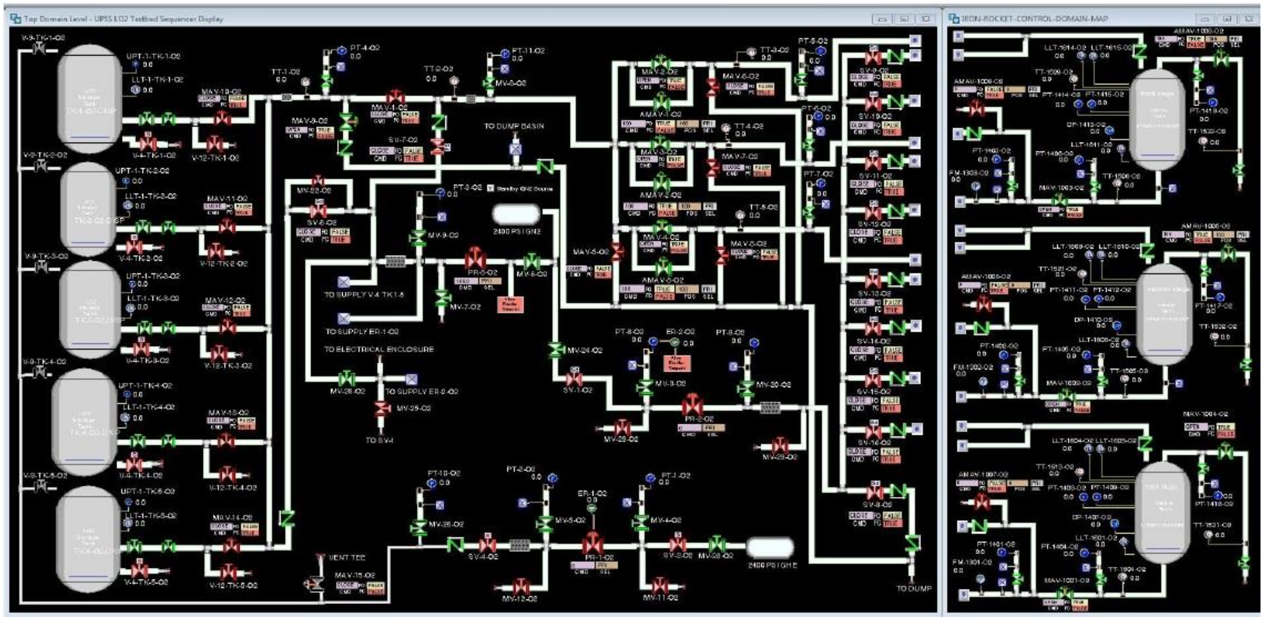


Figure 9. APL UPSS and Iron Rocket Domain Maps

4. Verification and Validation

One of the biggest challenges experienced in the deployment of autonomous operations software is that of verifying that the system has been built and performs according to requirements. While requirements verification of any system is always a challenge, autonomous operations are particularly challenging because the system is designed to perform response actions without supervision. For PHM enabled autonomy, the system has the benefit of providing information regarding its assessed state, and therefore can be checked and verified according to what has been inferred or correlated. In other words, the PHM assessment can be verified independently from the AO system's response actions.

A similar situation exists for the verification of diagnostic logic. Event driven fault diagnosis, root cause isolation, and downstream event propagation can be verified based on a simulation of detected events, thereby decoupling the verification of the fault models from the event detection. Event detection typically involves algorithms applied to sensor data and the verification of event detection algorithms does not need to concern itself with the details of the diagnoses.

Requirements verification of the AO system for APL also included the creation of repeatable unit and functional tests for all software modules. These tests were written in a test framework for G2 that conforms to industry standards for unit testing. The test cases were defined according to user story descriptions of the software requirements and therefore represent a suite of software quality assurance tests that can be used for ongoing regression testing of the AOS software throughout its lifecycle.

Validation of APL using Fault Injection Through Simulation

In the case of APL, the AO system has been specified to complete its mission of cryogenic propellant loading using the specified and executing sequence plan. Monitors are configured in the plan that, when activated, will trigger autonomous safing of the system. Also included are checks to verify that the pressures and temperatures of the commodity within the loading system are indicative of a liquid state. If the system fails to remain sufficiently chilled, the cryogenic commodity will boil off to vapor, thereby inhibiting flow and introducing both inefficiency and safety risk.

To aid in the validation of the AO for APL system even before the APL hardware had been constructed (refer to Figure 8), the project called for the implementation of a high fidelity physics model to be employed for driving the telemetry. This model, developed by team members from Stennis Space Center (SSC), provided a means by which to test the system end-to-end even before any real sensor data was being acquired. The model representation for the APL physics based simulation model is shown in Figure 10.

The physics model based simulator not only provided a means by which to drive system wide sensor data into the AO platform, but it also provided a means by which to simulate anomalies. Anomaly insertion using the Stennis developed simulator involved some handshaking between the simulator and the sending of AO valve commands. Sent commands were either used to advance the simulator according to a different configuration, or they would be used to trigger specific sensor responses based on the expected impact of the commands to the system.

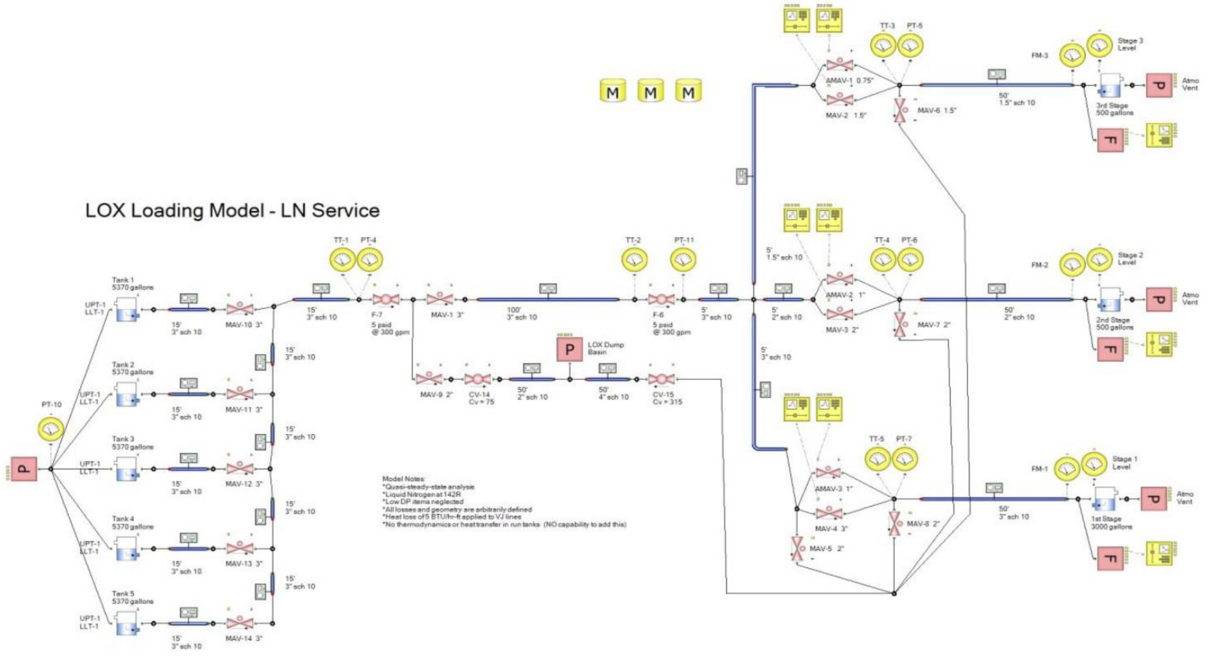


Figure 10. APL UPSS System Simulation Model

Of particular interest during the V&V of the APL system was the fact that the set of target failure modes were specified by domain experts to be detected based on component specific criteria – thereby resulting in a large table of detailed "possible" faults along with their corresponding (and often ambiguous) fault signatures. By using a PHM enabled AO system, it was possible to model the entire set of possible faults using a small set of generic fault models. These generic fault models would then be employed whenever any of the corresponding generic events were detected from simulated sensor data.

All validation testing was performed by conducting otherwise nominal loading using the AO sequencer in synch with the physics simulation. Failures were introduced into the nominal runs without any knowledge concerning the timing (where in the loading sequence) or the "affected" components associated with the fault conditions. There were 15 variants of the simulation run in order to present all scenarios, and the results were satisfying. Detailed test results of some failure modes can be found in [20]. As can be seen in Table 1, all but two of the categories of faults were reliably detected regardless of which failures were introduced or when. The significance of the validation results is amplified when it is recognized that the action response requires understanding of operational context and the accurate distinction between actual valve faults and position indicator faults.

Table 1. Validation Results

Failure Modes Injected	Response Validated
Valve Position Sensor Failure for 1st, 2nd, or 3rd stage Vehicle Fill and Drain Valve	Yes
UPSS vehicle skid loading valve stuck fully open	Yes
MAV2, MAV3, or MAV4 starts to close but does not fully close	Yes
Unsolicited valve transition (uncommanded valve movement)	Yes
Valve not Following Command	Yes
Detection of pressure leaks in isolated subsystems	Yes
Conflicting valve commands between primary and backup	Yes
Low Pneumatic system pressure during terminal countdown during pre-pressurization for flight	Yes
Vent Valve on flight tank is not seated and leaking during first half of fast fill on any of the 3 stages	Yes
Main pneumatic solenoid valve fails closed during any phase	No
Common mode instrumentation failure (pressure, temperature)	Yes
Loss of power on bus during load, replenish, drain	No

The two failure cases that were not properly verified using the Stennis Simulator for APL were a common mode failure of one of the power buses (not simulated), and the detection of a pneumatic system solenoid valve failure (insufficient sensor information on pneumatic supply, no position feedback sensor on APL solenoid valves).

5. SUMMARY

When architected as part of an overall model-based approach for delivering autonomous operations of mission and safety critical systems, it has been demonstrated that the technological capabilities of PHM systems can be a key enabler for AO software. Challenges to success include the complexities in the design and development of AO software, the lack of generic, modeling paradigms that support reliability focused design and facilitate context based reasoning, execution platforms that can be tightly configured to ensure timely real-time performance and assure deterministic responses, and the inability to leverage software development tools and processes that are designed to minimize the cost of delivering high quality software.

The accomplishments/results from this effort are summarized below [20]:

(1) Verification and Validation for Autonomous Operations achieved by using physics models and simulator

(2) Application development support for real-time ground support equipment for cryogenic propellant commodity (both LO2 and LCH4)

(3) Results demonstrate support for mitigation procedures that allows safe continuation of operations when conditions are satisfied, or autonomous safing of system when they are not

(4) NASA Technology Readiness Level (TRL) from validation in laboratory environments (TRL 4) to validation in relavent environments (TRL 5).

Future work includes additional prototypes for taking the system to TRL 6 (prototype demonstration in relevant ground/space environments) and TRL 7 (system prototype demonstration in space environment).

ACKNOWLEDGEMENTS

The authors thank Jay Gurecki and Gerald Stahl of NASA Kennedy Space Center, Lauren Underwood of NASA Stennis Space Center, and Kim Wilkins of General Atomics for their ongoing support of this work, along with their dedication to advancing the state of the art of Autonomous Operations.

REFERENCES

[1] NASA Technical Information Content web site: www.nasa.gov/content/automated-system-to-help-make-propellant-loading-more-efficient

[2] 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.

[3] P. Schmidt, D. Brueck, M. Rokey, and J. Pope. “Independent Assessment of Two NASA Fault Management Software Architectures”. NASA Fault Management Workshop, New Orleans, LA, Aug. 2012.

[4] J. Medina, J. Sass, J. Youney, and W. Schmitz. “Simulated Propellant Loading System: Testbed for cryogenic component and control systems research and development”. IOP Conference Series: Materials Science and Engineering 101, 2015.

[5] R. Schweickart. “Thermodynamic analysis of a demonstration concept for the long-duration storage and transfer of cryogenic propellants”. Cryogenics, Vol 52, 2014

[6] A. Majumdar, T. Steadman, J. Maroney, J. Sass, and J. Fesmire. “Numeric Modeling of Propellant Boil-Off in a Cryogenic Storage Tank”. Cryogenic Engineering Conference, Chattanooga, TN, 2007.

[7] J. Moubray, Reliability Centered Maintenance, Second Edition. New York, NY: Industrial Press, 1997, pp. 123–135.

[8] M. Schwabacher, K. Goebel. “A Survey of Artificial Intelligence for Prognostics”. AI for Prognostics Symposium, Arlington VA, 2007.

[9] B. H.Stamatis, Failure Mode and Effect Analysis: FMEA from theory to execution. Milwaukee, WI: American Society for Quality, 2003.

[10] R. Kapadia. “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

[11] 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.

[12] R. Kapadia, R. Gross, M. Walker, M. Venkatesh. “Health Monitoring Assessment and Prognostics (HealthMAP™) for Advanced Arresting Gear System. PHM Society Conference 2009, San Diego, CA, October 2009.

[13] M. Walker, “Model-based Reasoning Applications for Remote Intelligent Systems Health Management”, ASNE Intelligent Ships Symposium, May 2007.

[14] M. Walker, “Next Generation Prognostics and Health Management for Unmanned Vehicles”, Proceedings of the IEEE Aerospace Conference, Big Sky, MO, March 2010.

[15] 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, December 2007.

[16] M. Walker, R. Kapadia “Integrated Design of Online Health and Prognostics Management”, PHM Society Conference 2009, San Diego, CA, October 2009.

[17] Department of Defense. “The role of autonomy in DoD systems”, Defense Science Board, DoD, Washington D.C. 2012.

[18] U.S. Air Force. “Autonomy science and technology strategy”, U.S. Air Force Research Lab, Dayton, OH, 2013.

[19] M. Endsley, D. Jones. “Designing for situation awareness: An approach to human-centered design”, 2nd edition, Taylor & Francis, London 2012.

[20] Toro Medina J., Wilkins K., Walker M., Stahl G. “Autonomous Operations System: Development and Application”, Annual Conference of the Prognostics and Health Management Society 2016, Denver, CO, October 2-16.

[21] Toro Medina J., Figueroa F., Wilkins K., Walker M., Stahl G., Dito S., “Integrated System for Health Management and Autonomous Control (ISHM-AC) for Cryogenic Operations on the Simulated Propellant Loading System”, 52nd AIAA/SAE/ASEE Joint Propulsion Conference, Salt Lake City, UT, July 2016

[22] Toro Medina J., “Autonomous Operations Mission Development Suite”, Gensym User Forum, Orlando, FL, September 2016

BIOGRAPHY

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 four patents in the field. His work with HUMS and PHM began with BFGoodrich Aerospace, Vergennes, VT in 1996. He also spent 6 years as Senior Consulting Engineer for expert system manufacturer Gensym Corporation. He spent 10 years applying PHM with General Atomics, and founded D2K Technologies in 2014. He resides with his family in Oceanside, California.*

Fernando Figueroa *received his BS Degree at the University of Hartford, CT (1983), and MS and Ph.D. Degrees at The Pennsylvania State University, PA (1984, 1988); all in Mechanical Engineering. He was faculty of Mechanical Engineering at Tulane University (New Orleans) for 10 years, and Associate Chair of Advanced Instrumentation and Control at The University of New Brunswick (Canada) for 2 years. He is at NASA Stennis Space Center since 2000. He has lead multiple R&D projects in collaboration with academia, industry, government agencies, and NASA. He is currently Lead for Autonomous Systems and Operations, working on projects funded by NASA Advanced Exploration Systems, Space Technology Mission Directorate, and NASA Stennis Space Center. His areas of interest include Autonomous Operations and Systems, Integrated System Health Management (ISHM), Intelligent Systems, Intelligent Sensors, Robotics, and Automatic Controls.*

Jaime A. Toro Medina *Obtained his BSME (2008) and MSME (2010) degrees at the University of Puerto Rico Mayaguez Campus (2008) where he worked in the field of Computational Fluids Dynamics. Jaime also obtained an MS in Aerospace Engineering (2013) from the Georgia Institute of Technology where he worked as a research assistant on space system research. After a summer internship, Jaime joined the NASA KSC work force as a research and development engineer at the Cryogenics Test Laboratory. Currently, Jaime is performing as a technical project lead and software developer of autonomous systems for the Autonomous Propellant Loading project. In his spare time, Jaime enjoys salsa dancing, and lives with his family in Orlando, FL.*