Pre-Launch Diagnostics for Launch Vehicles[1,2]

Mark Schwabacher
NASA Ames Research Center
Mail Stop 269-3
Moffett Field, CA 94035
650-604-4274
mark.a.schwabacher@nasa.gov

Robert Waterman
NASA John F. Kennedy Space Center
Mail Stop LX-E
Kennedy Space Center, FL 32899
321-867-6680
robert.d.waterman@nasa.gov

Abstract—This paper discusses the opportunities for automating pre-launch fault detection and diagnostics for launch vehicles, by using ground-based computers to analyze data obtained from the pre-launch umbilical during integrated test and launch countdown operations in near-real time. Automation will improve the time to diagnose and isolate failures, which is critical for planned space missions that require multiple coordinated launches. The paper describes a prototype diagnostic system that we are currently developing for Ares I-X and the foundation that will be developed along with the prototype that will lead to certification of a diagnostic system. The prototype diagnostic system will focus on monitoring and diagnosing the Ares I-X first-stage thrust-vector control system and solid rocket motor during pre-launch activities at Kennedy Space Center. It will analyze data from existing sensors using rule-based, model-based, and data-driven algorithms.

TABLE OF CONTENTS

1. INTRODUCTION

This paper discusses the opportunities for automating pre-launch fault detection and diagnostics for launch vehicles, by using ground-based computers to analyze data obtained from the pre-launch umbilical during integrated test and launch countdown operations in near-real time. Automation will improve the time to diagnose and isolate failures down to a Line Replaceable Unit (LRU) that can be quickly removed and replaced. Use of an automated diagnostic system will also be evaluated to support retest of the new LRU after installation.

The paper will describe a prototype diagnostic system that we are currently developing for the first test flight of the NASA Constellation program, named Ares I-X, and the foundation that will be developed along with the prototype that will lead to certification of a diagnostic system. The National Aeronautics and Space Administration (NASA) requires that computational systems providing data used for making critical decisions regarding human spaceflight must be certified. Systems that are not certified can be utilized as advisory systems, however any data provided must be validated using a certified system. Due to this requirement, the position of the Constellation Ground Operations Project is that any diagnostic system developed must be certified.

NASA's current plan for returning humans to the moon is known as Project Constellation. It will use two launch vehicles. First the uninhabited Ares V Cargo Launch Vehicle [1] will launch the Lunar Surface Access Module (LSAM) and the Earth Departure Stage (EDS) into Low-Earth Orbit (LEO). A short time later, the Ares I Crew Launch Vehicle [2] will launch the Orion capsule containing four astronauts into LEO, where it will dock with the LSAM and EDS. Figure 1 shows Ares I and Ares V, and compares them with the current Space Shuttle and the Apollo-era Saturn V. It will be extremely important to avoid delays in the launch of Ares I, since the LSAM and EDS will only be able to survive in LEO for a limited period of time. Delays in the launch of Ares I could result in loss of mission. Historically, 46% of Space Shuttle launches have been delayed by a day or more after the start of the

[1] U.S. Government work not protected by U.S. copyright.

[2] IEEEAC paper #1063, Version 2, Updated December 14, 2007

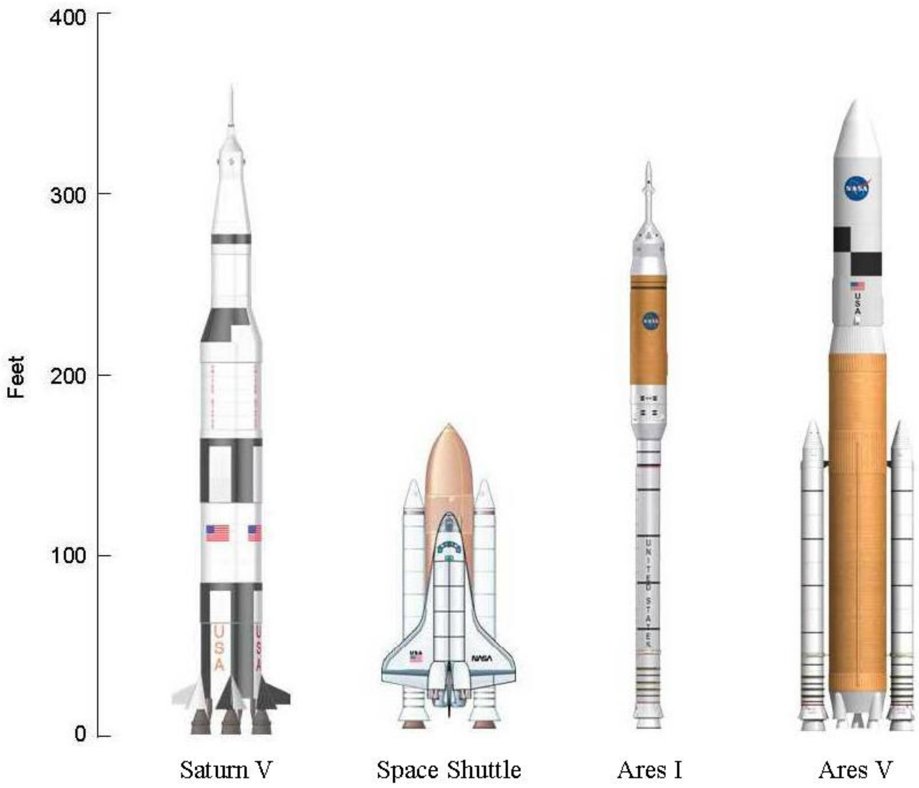

Figure 1 – The Saturn V, Space Shuttle, Ares I, and Ares V launch vehicles, shown to scale

two-day countdown, and the majority of these delays has been caused by hardware problems (while a minority of the delays has been caused by weather) [3]. Reduced turnaround time from fault detection to retest is critical to meet Launch Availability requirements.

Ares I-X [4] will be the first uninhabited test flight of Ares I. It will have a non-functional second stage and capsule with simulated mass and outer mold lines similar to those of the Ares I vehicle. The first stage will have a four-segment Solid Rocket Motor (SRM) with an inert fifth segment. (The Space Shuttle uses two four-segment SRMs, and Ares I will use a five-segment SRM.) Ares I-X is scheduled to launch in April of 2009. Our prototype diagnostic system will focus on monitoring and diagnosing the Ares I-X first-stage thrust-vector control (TVC) system and SRM during pre-launch activities at NASA Kennedy Space Center (KSC), including vehicle assembly in the Vehicle Assembly Building (VAB) and pre-launch activities at the launch pad. We may extend the scope to include the assembly and processing operations that occur in the Assembly and Refurbishment Facility.

We selected the TVC and SRM systems to model because they are systems that will exist on Ares I-X and that will be similar to the systems that will eventually be used on Ares I. An eventual Ares I ground diagnostic system based on our prototype would probably model several more systems (including the second-stage systems). The Ares I systems to include in the ground diagnostic system would be selected based on their likelihood of contributing to a launch delay.

The prototype will analyze data from existing sensors using rule-based, model-based, and data-driven algorithms. Rule-based algorithms encode expert knowledge in rules that are used to diagnose the system by mapping symptoms to failure modes. Model-based diagnostic algorithms use a hand-built hierarchical model of the system that describes how the system should work and its failure modes. The rules or models are verified using data, and eventually used to process near-real-time data. Data-driven algorithms differ from rule-based and model-based algorithms in that they automatically construct a model from the data. Sections 3-5 of this paper describe the three specific tools that we have chosen for the prototype, TEAMS (which is model-based), IMS (which is data-driven), and SHINE (which is rule-based).

The prototype will serve to test a particular combination of hardware and software that will be considered for use in future ground diagnostic applications. The prototype, together with a separate prototype focusing on the hydraulic Ground Support Equipment (GSE), will hopefully be used by the Constellation Ground Operations Project to determine if this approach is viable for diagnosing GSE anomalies, or if a more traditional approach using subsystem-based FDIR (Fault Detection, Isolation, and Response) should be employed.

Previous attempts to implement diagnostic tools or similar health management ground systems in support of the Space Shuttle program have ultimately produced uncertified advisory tools that could only be utilized in the support room. One of the main tenets of the ground diagnostic system for Ares I will be to consider certification as a primary goal from the beginning. Through following documented standard development processes, and evaluating key technologies for certification prior to implementation, the outcome of a certified ground diagnostic system with which operations engineers in the prime firing room can make launch “Go/No Go” decisions can become a reality. The paper will discuss some of the challenges faced by an automated pre-launch diagnostic system, including the path to certification and the lack of data for a new vehicle such as Ares I-X.

2. DEFINITIONS

Many of the words, phrases, and acronyms that are used in the field of Integrated Systems Health Management (ISHM) are ill-defined, so we begin with some definitions. First of all, we consider several different names for the field to be synonymous, including ISHM, Integrated Vehicle Health Management (IVHM), Fault Detection, Isolation, and Response (FDIR), and Fault Detection, Diagnostics, and Response (FDDR). *Fault detection* is detecting that something is wrong. *Fault isolation* is determining the location of the problem. *Diagnostics* is determining the specific failure mode. *Prognostics* is detecting the precursors of a failure, and predicting how much time remains before a likely failure. All of these methods take as input the steam of sensor values and commands, and output assessments of the system’s health. Our plan is for the Ares I-X Ground Diagnostic Prototype to perform fault detection, fault isolation, and diagnostics. We may consider adding prognostics to future systems.

3. MODEL-BASED DIAGNOSTICS

Model-based diagnostic algorithms encode human knowledge via a hand-coded representation of the system. Such a model can be physics-based (encapsulating first principles knowledge using systems of differential equations, for example), or can use other representations such as hierarchical declarative models. Some model-based techniques use a hierarchical model of the system with finite-state machines as the component models, including Livingstone [5, 6], MEXEC [7], and Titan [8]. HyDE (Hybrid Diagnosis Engine) [9] extends the Livingstone approach by also supporting systems of numerical equations as the component models. In systems based on Qualitative Reasoning [10], a hand-coded model uses qualitative, rather than numerical, variables to describe the physics of the system.

KATE (Knowledge-based Autonomous Test Engineer) is a model-based diagnostic system developed at NASA KSC [11, 12, 13]. It uses numerical simulations of the system being monitored, and signals a fault when the real-time sensor values fail to match the simulated values. It has object-oriented models of the system’s fault modes. After it detects a fault, it forms fault hypotheses based on the fault models, and then simulates each hypothesized fault. When it finds that the simulated data from one of simulated faults matches the data from the real fault, it reports the corresponding hypothesized fault as its diagnosis. KATE has been demonstrated for monitoring and diagnosing several NASA systems, including the Space Shuttle cryogenic tanking system, the Space Shuttle environmental control and life support system, and the Space Station power management and distribution system.

TEAMS (Testability Engineering and Maintenance System) is a commercial product from Qualtech Systems Inc. [14] that was originally developed using NASA SBIR (Small Business Innovative Research) funding. It uses a hierarchical model of the system and a qualitative model of failure propagation. It includes a design tool known as TEAMS Designer that can be used to create TEAMS models, and a real-time diagnostic tool known as TEAMS-RT that uses the TEAMS models to diagnose faults. TEAMS has been used for helicopters and for aircraft engines. TEAMS is currently being used at NASA Ames Research Center (ARC) to model the entire Ares I vehicle in support of three purposes. The first purpose is to support sensor placement through ambiguity group analysis. The second purpose is to model failure propagation time in order to help define the conditions under which a crew abort would be initiated. The third purpose is for use in an eventual ground diagnostic system for Ares I. All three of these purposes require the use of a common model. For the first two purposes, TEAMS is being used as a design tool only, and will not be flown or certified. In parallel with these activities at ARC, Honeywell is currently using TEAMS to model the Orion Crew Exploration Vehicle capsule. The plan is for the resulting model and the TEAMS-RT diagnostic system to be certified and flown on Orion.

We decided to use TEAMS as our model-based tool so that we could take advantage of the Ares I models that are already being developed at ARC, and also take advantage of the certification process that will be performed for Orion. We also feel that TEAMS models are simpler than those used by some other model-based diagnostic systems, such as HyDE, making it easier to certify TEAMS-RT and the TEAMS models.

4. DATA-DRIVEN FAULT DETECTION

Anomaly detection algorithms, also known as outlier detection algorithms, seek to find portions of a data set that are somehow different from the rest of the data set. A supervised anomaly detection algorithm requires training data consisting of a set of examples of anomalies, and a set of examples of non-anomalous (or nominal) data. From the data, the algorithm learns a model that distinguishes between the nominal and the anomalous data points. Supervised anomaly detection algorithms typically require tens or hundreds of labeled examples of anomalies, plus a similar number of labeled examples of nominal data points, in order to obtain adequate performance. Some examples of supervised leaning algorithms include artificial neural networks, decision trees, and support vector machines (SVMs).

Unsupervised anomaly detection algorithms are trained using only nominal data. They learn a model of the nominal data, and signal an anomaly when new data fails to match the model. They typically require tens or hundreds of nominal data points in order to obtain adequate performance.

We have been using historical data from the Space Shuttle Solid Rocket Boosters (SRB) to train and test the fault detection algorithms, since the Ares I-X first stage will be very similar to the Space Shuttle SRB (see Section 8). There have been very few significant anomalies in the Space Shuttle SRB. We therefore decided to use unsupervised anomaly detection algorithms, since they do not require labeled examples of anomalies.

We previously applied several unsupervised anomaly detection algorithms to historical data from the Space Shuttle Main Engine [15, 16]. These algorithms included Orca [17], which uses a nearest-neighbor approach, defining a point to be an anomaly if its nearest neighbors in feature space are far away from it, GritBot, a commercial product from RuleQuest Research [18], one-class support vector machines, Gaussian Mixture Models, Linear Dynamic Systems, and the Inductive Monitoring System (IMS).

The Inductive Monitoring System, developed by David Iverson at NASA ARC [19], is similar to Orca in that it is distance-based. The major difference is that during the training step, it clusters the nominal training data into clusters representing different modes of the system. At run time, it uses the distance to the nearest cluster as an anomaly measure. IMS is currently running on a console at the Mission Control Center at Johnson Space Center in uncertified mode to find anomalies in near-real-time data from the Control Moment Gyroscopes on the International Space Station (ISS). NASA is in the process of certifying IMS for use in this application.

We considered all of these unsupervised anomaly detection algorithms for use with Ares I-X. We have tentatively selected IMS as our unsupervised anomaly detection algorithm, primarily because it will allow us to leverage the certification effort that is already underway for ISS. We also feel that IMS is simpler and easier to understand than some of the other unsupervised anomaly detection algorithms, such as neural nets or one-class SVMs, which we believe will make it easier to certify.

5. RULE-BASED DIAGNOSTICS

Rule-based expert systems encode human knowledge in a set of rules written in a special-purpose language. An inference engine determines which rules are applicable and executes them at the appropriate times. When used for diagnostics, the rules typically map fault signatures to fault modes. Automated diagnostic systems are also often coded in conventional programming languages such as C. In such conventional programming languages, if-then statements can be used to represent knowledge about the mapping from fault signatures to fault modes.

Two rule-based expert systems that have been used for automated diagnostics are SHINE (Spacecraft Health Inference Engine), which was developed at the Jet Propulsion Laboratory [20], and G2, which is a commercial product [21]. G2 provides many features that have been built on top of the expert system, including a graphical user interface, and both model-based and data-driven diagnostic capabilities. Because G2 has so many features, it is very complex and has heavy computational requirements. SHINE by contrast is very simple and very fast. SHINE uses a data-flow representation to execute rules efficiently, which enables it to execute over 300 million rules per second on current desktop computers.

SHINE has been used for ground operations for several missions, including Voyager and the Extreme Ultraviolet Explorer. It has been tested on flight hardware (the X-33 Avionics Flight Experiment) and in flight (on a NASA Dryden Flight Research Center F/A-18). We have tentatively selected SHINE for use in our prototype because of its speed, because of its flight heritage, and because we believe that its simplicity will make is easier to certify.

6. PROTOTYPE DEPLOYMENT PLAN

We are hoping to deploy our prototype to Hangar AE at Kennedy Space Center. Hangar AE is one of the "back rooms" in which engineers monitor data from a spacecraft and from the ground support equipment during the pre-launch phase. For Ares I-X, the prime launch team, consisting of twenty operations engineers and test directors, will be in the launch control center (LCC) in the Launch

Complex 39 area. The engineering and Safety & Mission Assurance (S&MA) support team, as well as public affairs, will all be in Hanger AE. Currently, all of the vehicle data from launches of Atlas and Delta expendable launch vehicles is delivered in near-real-time to Hangar AE. The plan is for all of the data related to the Ares I-X launch to also be delivered to Hangar AE, including data from the launch vehicle, Ground Support Equipment (GSE), and ground command and control systems, and including data both from the launch pad and from the VAB. Currently the data is streamed from a server known as the Winplot Archive Server to a client application known as Winplot. Winplot then plots the near-real-time data so that the engineers can examine it. We plan to interface our prototype with the Winplot Archive Server so that it can obtain near-real-time data from the server. In addition to Winplot, the Hanger AE system supports a graphical display environment called IRIS that uses Web-based monitor-only displays. Outputs from the Ground Diagnostic Prototype will be displayed using the IRIS system.

The current plan is for our software to run on a separate computer in Hangar AE. This computer will display the outputs of the three tools that we have selected. For displaying the outputs from TEAMS-RT and SHINE, we plan to develop displays within the IRIS graphical display environment. We plan to use the Winplot tool to display the IMS "anomaly score" outputs. We have not yet decided whether to build a tool that would combine the outputs of the three algorithms into a single unified diagnosis. (Such a tool would correlate faults detected by IMS with diagnoses from TEAMS-RT and SHINE.) We might also decide to use the output of IMS – its "anomaly score" – as an input to TEAMS-RT and/or SHINE.

We have tentatively selected TEAMS-RT, IMS, and SHINE for use in our prototype. We may reconsider this selection as we learn more about the performance of each algorithm on relevant data. We intend to test several other algorithms at ARC (without deploying them to KSC or interfacing them with the Winplot Archive Server). These other algorithms may include the HyDE diagnostic algorithm and several unsupervised anomaly detection algorithms (Orca, one-class SVMs, GritBot, Gaussian Mixture Models, and Linear Dynamic Systems). If we find that one of these other algorithms performs much better than the algorithms that we have tentatively selected, then we may decide to incorporate one of them into the prototype.

Our prototype is intended to be a prototype of the pre-launch diagnostic system that will eventually be built for Ares I. If the prototype is successful, then we expect that some of the technology from the prototype would be migrated into the Ares I production diagnostic system.

7. CERTIFICATION

Certification is a major challenge for software of the type that we are using in our prototype. As far as we know, the three types of software that we are planning to use have never before been certified for use in human spaceflight. One of the main goals of this work is to determine how difficult it would be to certify this type of software to be used for Ares I. Since Ares I-X will be uninhabited, it will not have the same certification requirements as Ares I – its certification requirements will be more like those for expendable launch vehicles. Some of the certification challenges include applying NASA's certification requirements for Constellation against previously developed COTS or GOTS software. Certification includes more than just verification and validation; it also includes development practices, documentation, configuration management, and pedigree management of all the components that go into the system which is seeking certification.

We will need to establish a set of requirements for the prototype against which we will attempt to certify it. These requirements will include a list of failure modes to be detected and diagnosed, several requirements on accuracy, such as the missed detection rate, the false alarm rate, and the misdiagnosis rate, and a throughput requirement specifying the number of samples per second that must be processed.

One challenge we will face in certifying model-based systems such as TEAMS and rule-based systems such as SHINE is deciding whether to certify the input to the special-purpose compiler or the output of the compiler. In particular, we are building a TEAMS model of the vehicle. TEAMS compiles this model into a form known as the *D*-matrix. The runtime system known as TEAMS-RT interprets the *D*-matrix at runtime in order to detect and diagnose faults. We can either certify the TEAMS model and the TEAMS compiler, or we can certify the *D*-matrix. (The certification of the *D*-matrix would ensure that the true-positive, true-negative, false-positive and false-negative rates are acceptable.) In either case, we will also need to certify the TEAMS-RT runtime system (including certifying the code that determines test results). Similarly, SHINE compiles a set of rules into C code. We can either certify the rules and the SHINE compiler, or we can certify the resulting C code. In both cases, we believe it will be easier to certify the output of the compiler (the *D*-matrix and the C code), but we still need to explore these questions further.

Another decision that we will need to make regarding certification is the decision about which portions of the tools will be certified by test, by analysis, or by a combination of test and analysis. We expect that different parts of different tools will be amenable to different certification methods.

We have set two certification-related goals for ourselves. The first is to attempt to get our prototype certified for Ares I-X. If we achieve that goal, then the engineers in Hangar AE will be able to make decisions based on the outputs of our prototype. If we do not succeed in getting the prototype certified, then we plan to run the prototype in Hangar AE in uncertified mode. In the latter case, engineers will need to use other certified software to verify the outputs of the prototype before making decisions based on the outputs. The process of attempting to get the prototype certified for Ares I-X, and the success or failure of that attempt, will teach us something about the certifiability of the types of algorithms that will be included in the prototype.

Our second certification-related goal is to assess how difficult it would be to certify our prototype for human spaceflight. We plan to review the certification requirements for Ares I as they become available in order to help us make this assessment.

8. Data

Because Ares I-X is a new vehicle for which test and flight data do not yet exist, we plan to use historical data from the Space Shuttle to train IMS and to test all three algorithms in the short term. The first stage of Ares I-X will be very similar to the Solid Rocket Boosters (SRBs) on the Space Shuttle, using legacy hardware for the aft skirt including the TVC system with a repackaged avionics control system. Due to these similarities in hardware systems we believe that the data from the first stage of Ares I-X will be very similar to the data from the Space Shuttle SRBs with respect to the hydro-mechanical aspects of the TVC system. We have therefore decided to use historical Space Shuttle SRB data to train and test our algorithms. Eventually we may also use simulated Ares I-X data and data from tests of Ares I-X components as they become available. The large amount of potentially relevant training data and the volume of data in the near-real-time data stream data may prove to be challenges for some of the algorithms that we are testing.

9. Current Status

We have obtained SRB data from two Space Shuttle flights, covering the final three hours before launch. These data contain several hundred parameters representing the entire SRB and some of the related ground support equipment. These data do not include vehicle assembly. Eventually we plan to obtain data that covers a much longer period of time for each flight, including vehicle assembly, and we plan to obtain data for several more flights, but we have started our analysis with this limited set of data.

We have obtained the TEAMS Designer model of the Ares I first-stage TVC, and have started to adapt it for diagnostic use for Ares I-X by adding test points. We have installed the Winplot Archive Server software onto a server at NASA ARC, and have begun to interface TEAMS-RT and IMS with it. We have also obtained space within Hanger AE during the Ares I-X mission to house the ground diagnostic prototype.

Based in part on the TEAMS model of the Ares I TVC, we selected a subset of the Space Shuttle SRB data set that is most relevant to the TVC. This subset contains fewer than one hundred parameters. We are currently in the process of applying IMS to this subset of the data.

10. Conclusions

Automated pre-launch diagnostics offers the potential to reduce delays in diagnosing problems in launch vehicles. Because of NASA’s plan to launch Ares I shortly after launching Ares V for an in-orbit rendezvous of the two spacecraft, reductions in launch delays are essential to mission success. A large part of reducing launch delays will be in isolating and diagnosing failures and then retesting the replaced components with minimal impact to the processing flow. Having an automated tool that is certified will allow operations engineers in the launch control firing room to effectively utilize the information without having to manually verify every output using time-consuming certified alternative methods.

We have begun to build a prototype automated pre-launch diagnostic system for Ares I-X. We hope that this prototype will demonstrate that automated pre-launch diagnostics can reduce delays significantly and can be certified for use in human spaceflight.

Acknowledgments

We thank all the members of the Ares I-X Ground Diagnostic Prototype team, including Rodney Martin and Barbara Brown of NASA ARC, Bob Ferrell, Richard Nelson, and Nathan Wood of NASA KSC, Ted Stodgell of NASA Marshall Space Flight Center, Peter Robinson, Charles Lee, and John Ossenfort of EASI, Lawrence Markosian of Perot Systems, Inc., and Rebecca Oostdyk of ASRC. We also thank Rodney Martin, Lawrence Markosian, and Peter Robinson for reviewing a draft of this paper. We thank David Iverson of NASA ARC for providing IMS. We thank Roland Lebon and Carol Thompson of United Space Alliance for providing Space Shuttle SRB data and expertise. This work is funded by the CLV Program within NASA’s Exploration Systems Mission Directorate.

REFERENCES

[1] "Constellation Program: America's Fleet of Next-Generation Launch Vehicles - The Ares V Cargo Launch Vehicle." NASA Fact Sheet FS-2006-07-84-MSFC, 2006. http://www.nasa.gov/pdf/151420main_aresV_factsheet.pdf

[2] "Constellation Program: America's Fleet of Next-Generation Launch Vehicles - The Ares I Crew Launch Vehicle." NASA Fact Sheet FS-2006-07-85-MSFC, 2006. http://www.nasa.gov/pdf/151419main_aresI_factsheet.pdf

[3] Cates, G. R., Cirillo, W. M., and Stromgren, C., "Low Earth Orbit Rendezvous Strategy for Lunar Missions," Proceedings of the 2006 Winter Simulation Conference, 2006.

[4] "Kennedy Prepares to Host Constellation", NASA Web page. http://www.nasa.gov/mission_pages/constellation/main/Constellationfacilities.html

[5] Williams, B. C. and Nayak, P. P., "A Model-based Approach to Reactive Self-Configuring Systems," Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA: AAAI, 1996.

[6] Kurien, J. and Nayak, P. P., "Back to the Future for Consistency-based Trajectory Tracking," Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA: American Institute for Artificial Intelligence, 2000.

[7] Barrett, A., "Model Compilation for Real-Time Planning and Diagnosis with Feedback," Proceedings on the International Joint Conference on Artificial Intelligence, 2005.

[8] Williams, B. C., Ingham, M., Chung, S., Elliott, P., and Hofbaur, M., "Model-based Programming of Fault-Aware Systems," AI Magazine, Fall 2003.

[9] Narasimhan, S. and Brownston, L., "HyDE – A General Framework for Stochastic and Hybrid Model-based Diagnosis," Proceedings of The 18th International Workshop on Principles of Diagnosis (DX-07), 2007.

[10] Weld, D. S., and de Kleer, J., Readings in Qualitative Reasoning About Physical Systems, San Francisco: Morgan Kaufmann, 1989.

[11] Parrish, C. L., and Brown, B., "Knowledge-based Autonomous Test Engineer (KATE)," The Second National Technology Transfer Conference and Exposition, NASA, 1991.

[12] Pepe, C., Beltz, S., Merchant, R., Nunez, S., and Goodrich, C, "KATE: From the Lab to the Firing Room," Florida Artificial Intelligence Research Symposium, 1994.

[13] Goodrich, C., "A Method for Diagnosing Time Dependent Faults Using Model-Based Reasoning Systems," Florida Artificial Intelligence Research Symposium, 1995.

[14] Qualtech Systems Inc. Web site. http://teamqsi.com/

[15] Schwabacher, M., Oza, N., and Matthews, B., "Unsupervised Anomaly Detection for Liquid-Fueled Rocket Propulsion Health Monitoring," Proceedings of the AIAA Infotech@Aerospace Conference, Reston, VA: American Institute for Aeronautics and Astronautics, Inc, 2007.

[16] Martin, R. A., Schwabacher, M., Oza, N., and Srivastava, A., "Comparison of Unsupervised Anomaly Detection Methods for Systems Health Management Using Space Shuttle Main Engine Data," Proceedings of the JANNAF Propulsion Meeting, 2007.

[17] Bay, S. D. and Schwabacher, M., "Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule," Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York: Association for Computing Machinery, 2003.

[18] RuleQuest Research Web site. http://rulequest.com/

[19] Iverson, D. L., "Inductive System Health Monitoring," Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, Vol. 2 & Proceedings of the International Conference on Machine Learning; Models, Technologies & Applications, MLMTA '04, June 21-24, 2004.

[20] James, M. and Atkinson, D., "Software for Development of Expert Systems," NASA Technology Briefs, vol. 14, no. 6, 1990.

[21] Gensym Web site. http://www.gensym.com/

BIOGRAPHY

Mark Schwabacher earned his Ph.D. in computer science in 1996 from Rutgers University. His thesis work applied artificial intelligence to engineering design. He has worked at NASA Ames Research Center for almost ten years, where he has worked on several ISHM activities. He served as the Software Lead of the NASA X-37 Integrated Vehicle Health Management Experiment, and is currently leading the development of the Ares I-X Ground Diagnostic Prototype in collaboration with NASA Kennedy Space Center, NASA Marshall Space Flight Center, and the Jet Propulsion Laboratory. He has also applied anomaly detection algorithms to Earth science and to aviation security.

Robert Waterman has twenty-one years experience in operational diagnostic and preventative maintenance of critical space avionics systems. He was the system specialist for the Space Shuttle Main Engine Avionics system. He was technical manager for real-time control application software of ground checkout and launch control system. He was the strategic technical manager for command, control and monitor systems and currently the Command, Control and Communications architect for the Constellation Ground Operations Project at the Kennedy Space Center. He has a Bachelor of Science in Computer Engineering from Florida Institute of Technology in 1985.