Model-Based Anomaly Management for Small Spacecraft Missions

Christopher Kitts Robotic Systems Laboratory, Santa Clara University 500 El Camino Real, Santa Clara CA 95053 408.554.4382 ckitts@scu.edu

Richard M. Rasay Robotic Systems Laboratory, Santa Clara University 500 El Camino Real, Santa Clara CA 95053; 408.551.6047 mikerasay@gmail.com

ABSTRACT

Space missions produce value through the production of mission data products and services. In doing so, however, significant resources are expended in order to maintain system health and to manage anomalies as they occur. These tasks are costly in terms of the expertise, personnel, and time required to detect, diagnose and resolve problems. Our recent work in model-based reasoning (MBR) techniques has demonstrated the applicability of this technology to the small satellite domain. MBR uses fundamental design knowledge of a system in order to compute reasoning conjectures relating to the existence of symptoms, diagnosis estimates, and resolution control actions. In doing so, it provides a systematic and efficient framework for automated reasoning, which in turn can dramatically accelerate the analysis of anomalies with significantly improved results. In this paper, we describe the MBR approach to anomaly management and review our theoretical and algorithmic contributions to this field. We outline our software toolboxes that implement these algorithms, and we highlight the tools that are being developed to apply this software to real space systems. Finally, we review results of using this reasoning system for several small satellite missions, ranging from the student-built Sapphire microsatellite to the NASA GeneSat-1 spacecraft.

1

INTRODUCTION

Historically, system anomalies have been managed through the use of human-based "experiential" reasoning techniques. Highly trained and experienced engineers embed their compartmentalized understanding, rules of thumb, intuitions, heuristics, and past experiences into a loose knowledge base composed of procedures, diagrams, handbooks, manuals, and remembered information.

Widespread reports in the space operations literature, as well as years of the authors' own experiences in operating a number of space systems, attest to the significant drawbacks of this approach. Human-based experiential systems suffer from high training and staffing costs, sensitivity to personnel changes, the impacts of human error, the inability to reuse knowledge and procedures across lifecycle phases and missions, the sensitivity of the knowledge base to small changes in the system, and many other factors ^{1,2,3}.

Together, these drawbacks can result in on-orbit operations costs that constitute 25-60% of overall mission lifecycle costs⁴; for the \$100 billion space industry, such system operations costs range in the tens of billions of dollars annually⁵. Declining federal outlays for space projects and increased market pressures on commercial space ventures are forcing the

space industry to lower these costs. As a result, new approaches for detecting, diagnosing and responding to system anomalies are of great interest.

Expert Systems

The first step in automating systems health management often consists of the implementation of an "expert system." More precisely, these systems are often production rule systems that process simple "rules" based on accumulated experience in operating the space system.

These expert systems have many benefits. As an automated tool, they are simple in concept and implementation, and they are able to execute established flight rules more precisely and considerably faster than human operators. The computational power required to execute them is minimal, making them viable even in embedded environments. The conclusions they draw are transparent in the sense that it is straightforward for a human to understand the rationale for conjectures made by or actions taken by the system^{6,7}.

Unfortunately, expert systems have numerous drawbacks. Computationally, they are limited given that their typical "if-then" rule construct limits sophisticated analysis. Furthermore, their reliance on

experiential knowledge leaves them susceptible to the drawbacks of an informal knowledge base that is often nothing more than a loose collection of facts with no consistency in detail or coverage⁸.

Even with these drawbacks, expert systems have been used successfully within the space industry in order to improve analysis and cut costs. The Magellan and Cassini missions both used expert systems, allowing rapid automated response when these probes were out of contact with the Earth; to enhance their reasoning capabilities, the systems were augmented with the ability to execute rules when the motivating condition persisted over time^{2,9}. The EUVE mission used an expert system in order to dramatically improve the costeffectiveness of ground-based health management¹⁰; this innovation culminated in a period of "lights-out" operation and was instrumental in extending the mission's operating life¹¹. Finally, the Sapphire microsatellite employed an on-board persistence-based production rule system that served to test several automation experiments⁸; one particular innovation was its use to inform a beacon-based health management system that was validated as a method for enabling cost-effect health management operations¹².

Model-Based Reasoning¹³

Model-based reasoning (MBR) is one of many alternate reasoning approaches for computing conjectures that can support the health management process. In MBR, reasoning conjectures are computed from fundamental design information regarding the design of the engineering system; as such, MBR is often defined as reasoning from first principles. For example, a system description is used to define the behavior of each component within a system and to declare the connectivity (e.g., the structure) of the components. With this information, the performance of the system can be modeled or simulated, thus allowing the predictions of output values given values for the system's inputs and an assumption that the system is operating nominally.

One of the most significant contributions in the field of MBR is a formal theory of fault detection and diagnosis^{14,15}. This theory defines a *fault* as a condition within a component that prevents it from performing in accordance with its explicitly defined (e.g., modeled) behavior. Stated formally, the definition of a behavior prescribes a specific value of a component's output signal given the values of its inputs; mathematically, the behavior is a constraint on the output value. *Detection* of a fault using an MBR technique is accomplished by comparing the outputs of the real system to the outputs of the modeled system; as shown

in Figure 1, an inconsistency between these values is interpreted as the *symptom* of a fault (given a number of assumptions, such as the accuracy of the model, etc.). A crucial distinction between MBR and other approaches to reasoning about faults is that MBR exploits models of proper functionality (which are developed extensively in the design phase of a system) rather than attempting to enumerate all possible failure modes.

Once the symptom of a fault has been detected, a diagnosis process isolates the specific components that may be faulted. The MBR approach to identifying these components relies on a process known as constraint relaxation. For a given component suspected of being faulty, its behavioral constraint is relaxed, which means that its output is permitted to range over its set of all valid output values. The system is then re-simulated in order to compute new values for the system's outputs. Depending on the nature of the system, the range of valid values from the possibly faulted component may lead to a range of possible values for one or more of the system's outputs. The observed values from the real system are compared to these new simulated values/ranges; if the observed values are members of the simulated sets of possible values, then the assumed component fault is, in fact, a valid diagnosis.

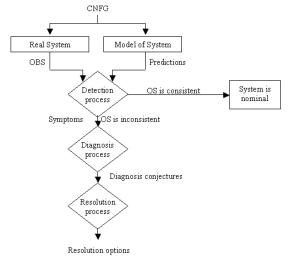


Figure 1: The Anomaly Management Process

MBR has been applied to fault detection and diagnosis in a variety of fields ranging from electronic circuitry ^{16,17} to spacecraft health management ¹⁸⁻²¹. These applications have motivated numerous extensions to the basic theory in order to incorporate empirical knowledge about failures ^{16,22,23}, to generate optimal sensing plans for efficient diagnosis ⁹, to address computational loading through the use of hierarchical

models and truth maintenance²⁴⁻²⁶, to incorporate timevarying behavior and observations into the analysis²⁷, and to address the task of fault recovery^{28,29}.

EXTENDED MBR FOR ANOMALY MANAGEMENT

Motivated by our work in operating numerous spacecraft ranging from military to university satellites, we have developed several conceptual extensions to MBR-based fault detection and diagnosis methods. These extensions have resulted in an extended MBR theory for anomaly management, a set of initial algorithms for executing this theory, and a toolbox of software that implements these algorithms.

The extended theory uses a fundamental view of a component. In this view, a component has inputs, outputs and internal states. The component implements one or more behaviors, and these behaviors drive the value of outputs as a function of inputs and states. The behaviors may only be valid under certain conditions, such as specific input or state operating ranges, and the design may call for associated constraints based on these conditions. Finally, connections link inputs and outputs to output and input ports on other components.

For as simple as this view is, a phenomenal level of information can be formally and systematically derived from it in the context of an operational multicomponent system. MBR provides the framework for achieving this through explicit behavioral modeling and the comparison of model outputs with real-world telemetry. Although the full theoretical framework is too extensive to detail here (see [30-32] for complete details), we provide a qualitative description of some of the highlights of the conceptual foundation.

Anomaly Types

We define three different types of anomalies:

- Fault: In keeping with the original theory, a fault is defined as the misbehavior of a component, as called for by the behavioral description of that component.
- <u>Hazard</u>: A hazard is a condition that violates an operating constraint that has been levied upon the design.
- <u>Misconfiguration</u>: A misconfiguration is an improper assumption regarding what the actual input configuration is.

The fundamental point of view that unifies these as anomalies is one of violated assumptions or mathematical constraints. With MBR, each aspect of the component definition provided in Figure 2 is

formally expressed through mathematical or logical representation. A behavior is an assumption, or constraint, on the value of a component output given the values of the associated inputs and states; a fault, therefore, as a condition that violates this assumption. Similarly, an operating condition is a constraint on component input or state values; a hazard is a violation of this type of constraint. And a configuration statement is an assumption regarding the value of a system input; a misconfiguration is a violation of this type of assumption.

All three of these types of anomalies occur regularly in the field operation of complex engineering systems such as space systems. These anomalies may exist independently of each other, and they may interact through causal relationships. All are potential threats to the health and performance of a system, and each has distinct implications regarding their effect and possible remedies. Being able to explicitly and formally detect, diagnose, and resolve each is fundamental to effectively controlling a space system in an efficient manner.

While these designations may seem straightforward, their definitions have heretofore not been used in a reasoning system that has formalized them in this manner nor which considers the relaxation of associated constraints as part of a diagnosis project. Furthermore, while anomalies such as misconfigurations may seem mundane due to their association with operator error and therefore unworthy of formal consideration, they play a significant role in the occurrence of anomalies; our unified framework allows them to be addressed as part of a coherent strategy of detecting, diagnosing and resolving an anomaly of any type.

Reasoning Framework 13

We have formalized our expanded set of operational anomalies, as well as a complementary suite of resolution actions, into a new, more comprehensive, model-based theory of anomaly management. Like the previous theory of fault detection and diagnosis, this conceptual foundation uses a consistency-based approach that identifies and resolves inconsistencies among assumptions in the model of the system and observations of the real system. Key elements of the theory include:

- The engineering system description, a collection of design-time model information indicating the systems structure, behavior, and intended use.
- The operational system description, a collection of operational-time model information that includes the engineering system, the intended application requirements, and real-time configuration and observation data.

- The definition of anomaly predicates for faults, hazards, and misconfigurations.
- The definition of resolution predicates for formally over-riding operating constraints and altering a mission application.
- Formal definitions for a detected *symptom*, a *diagnosis* conjecture, and a *resolution* action.

Management Tasks 13

Detection involves identifying inconsistencies between observations of the real system and the model of our system, to include a) the system's structure, behavior and intended operation, b) the intended application of the system, c) our belief that no anomalies exist, d) our initial policy of not over-riding any operating constraints or relaxing any mission requirements, and e) our understanding of the current configuration.

Diagnosis is also a reasoning process grounded in logical consistency. In effect, diagnosis implies finding all possible anomaly conjectures that relax an assumption in our model in order to re-establish consistency between the model and our observations. For example, assume that a fault or hazard for a particular component relaxes the component's output value; if this effect ripples through the system, as determined via simulation, in order to support the real observations as one possible effect, then the fault and hazard are both viable diagnoses.

The generation of resolution options continues the use of logical consistency in order to determine the options that re-establish consistency between the model, the observations, and the belief in a specific diagnosis. Resolution options include combinations of system reconfigurations (e.g., swap out a faulted component with its redundant unit, turn on the heater for an undertemperature component, etc.), constraint over-rides (e.g., explicitly accepting a violated component operating condition), and mission alterations (e.g., explicitly relaxing one or more system requirements).

ALGORITHMIC IMPLEMENTATION

We have developed reasoning algorithms to implement our extended MBR theory of anomaly management, and these algorithms have been prototyped as a Matlab toolbox for use in both test and operational environments. This software methodically computes symptoms, diagnoses and resolutions in accordance with the criteria specified in the theory's formal definitions.

Process Algorithms

The anomaly detection process is executed periodically, typically after telemetry acquisition and certainly in response to a command. For a given input configuration, the model is evaluated in order to predict system outputs. These outputs are then compared to telemetry from the actual space system. Specific checks identify conflicts with configuration assumptions, violations of intended component operating conditions, or inconsistencies with predicted observation outputs.

The diagnosis algorithm is invoked only when the detection algorithm identifies a symptom. As shown in Figure 2, the algorithm is implemented as a two-stage process given the appropriateness of two distinct decision-making approaches:

- In the first stage, a production rule process is used to directly identify misconfiguration and hazard diagnoses.
- In the second phase, symptoms generated by inconsistencies between the predictions and observations of the system state are addressed. A candidate generation algorithm produces a set of assumptions to evaluate, and these constraints are systematically relaxed to see if the resulting simulation re-establishes consistency with observations; if this occurs, the assumption is a possible diagnosis. To date, versions of this algorithm phase have been implemented in order to identify sets of diagnoses for multi-symptom single-remaining-anomaly cases.

The resolution process executes when the diagnosis algorithm returns a set of diagnosis conjectures. For a specific diagnosis, the algorithm is performed by alterations. systematically considering mission constraint overrides, and new configurations. The first two are easily generated: if any mission-critical system outputs or intended component conditions are violated, simply accepting these facts by relaxing the mission or overriding the constraints become resolution options. Of course, these are usually accepted only as a last Therefore, it is critical to evaluate reconfiguration options. This is done be re-evaluating the model for permutations of configuration options upstream of the diagnosis; new configuration combinations that result in the satisfaction of a violated mission requirement are saved as possible resolution options. Finally, the three types of resolution options are considered in systematic combinations; minimal combinations that re-establish consistency within the system (given the assumed diagnosis) are saved as valid resolutions.

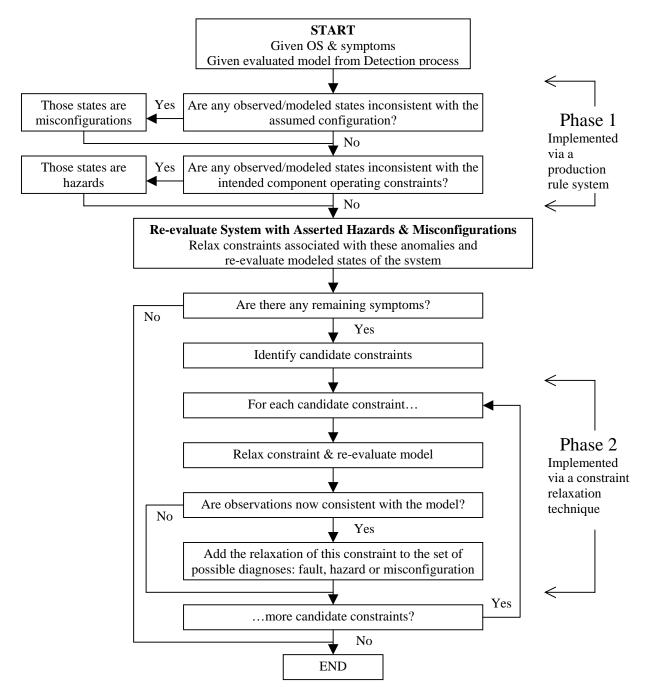


Figure 2: Simplified View of a 2-Phase Diagnosis Algorithm.

Reflections on MBR Processing

There are several characteristics of the anomaly management process worth highlighting. First, different types of anomalies can lead to the same symptom. This many-to-one correlation makes operational anomaly management a difficult task.

Second, the ability of an anomaly to be observed and to affect the rest of the system is a function of the system's configuration. Exploiting this dependence through computation is what ultimately allows MBR to be far more precise in its diagnosis and resolution conjectures compared to an experiential system, which often diagnoses problems using an "anything upstream" strategy.

Finally, because anomaly observability is configuration dependent, operational anomaly management is necessarily parsimonious; if there is no indication of an anomaly, then no anomaly is assumed given that mission critical systems cannot be arbitrarily reconfigured in order to verify a healthy state.

SMALL SATELLITE MISSION APPLICATION

We have experimentally verified and validated our anomaly management reasoning system by applying it to the operational control of the Sapphire and GeneSat space systems. In each case, this has been done in the context of configuration management for the end-to-end distributed space systems to include components in the mission control center, the remote communication stations, and on-board the satellites. A few examples are provided here.

To our knowledge, this is the most comprehensive application to date of MBR techniques to managing anomalies within space systems.

Sapphire

Launched in 2001, Sapphire was developed by students at Stanford University and supported a variety of missions to include digital communications, Earth photography, and sensor characterization³³. Faculty and students at Washington University in St. Louis prepared and integrated the satellite for launch, and primary control of the satellite was turned over to the U.S. Naval Academy to support student education. Faculty, staff and students at Santa Clara University operated the satellite through 2005 in order to conduct a series of research experiments involving automation technologies. In 2005, operations ceased do to the degraded state of the satellite's power system; the five years of operational life dramatically exceeded the one-year of planned design life.



Figure 3: Elements of the Sapphire Space System

For the experiments reported on here, the comprehensive Sapphire space system consisted of the Sapphire microsatellite, several automated communication stations located throughout the United

States, and a centralized mission control complex in the Center for Robotic Exploration and Space Technologies building in the NASA Ames Research Park, Moffett Field, California. Elements of this system are depicted in Figure 3. The communication stations were remotely controlled via the Internet by operators in the mission operations center, and amateur radio was used to communicate between these stations and the satellite when it was in view. The ground segment portion of this system, developed by students at Santa Clara University, is used to support the operation of a variety of other spacecraft and robotic missions^{34,35}.

Fortunately (or unfortunately, depending on the applicable point of view), several real and unanticipated anomalies involving the flight system were identified during these contacts. Combined with the many unplanned ground segment anomalies that occurred, these events provided a wonderful opportunity to exercise the anomaly management system in a realistic setting.

In all of the following cases, the anomaly management software successfully detected symptoms and generated a valid set of possible diagnoses (although on occasion this list was long and required active testing in order to narrow down the possibilities):

- Unresponsive operating system on the communication station computer.
- Misconfigured IP address on the communication station computer (the host institution changed the IP address without notifying the operations team).
- Power outage at the communication station facility.
- Out of date Keplerian elements used by the communication station autotrack software.
- Incorrect time on the communications computer resulting in inaccurate autotrack computations.
- Overheating of the transmitter amplifier leading to undesired performance.
- Misconfiguration of TNC settings by the operator.
- Misconfiguration of autotrack software settings by the operator.
- Failure of an antenna positioning servomotor (on several occasions).
- Improperly executed operations procedures resulting in misconfigurations.

One particularly interesting on-board anomaly involved the attempt to collect data from experimental infrared sensors that were one of the primary payloads on the Sapphire spacecraft¹³. During this procedure, the operator improperly executed the relevant command plan, failing to enable the sensors prior to collecting data thereby yielding a data set that was effectively garbled. When data was acquired from these parallel

sensors, it was inconsistent with the expected values thereby triggering an initial, automated diagnosis process that returned three possible diagnoses. Of these three, a possible resolution (short of altering the mission) existed for the "misconfigured sensor" conjecture. This resolution, to command the sensor to its enabled configuration, was executed and solved the problem. The speed of the diagnosis process, which executed in approximately 3.5 seconds, allowed the operator to properly configure the satellite, collect the desired sensor data, and complete other operational tasks within the 12-minute contact window.

For this example, it is instructive to consider the diagnoses that were not made. A simplified block diagram of the relevant on-board component configuration is shown in Figure 5 to illuminate this discussion. Diagnoses relating to component power were not made given that power was being properly supplied to other units within the satellite, as reflected by other telemetry that was consistent with simulated predictions. It is also interesting to note that faulty sensors were not a diagnosis. This is because the initial diagnosis that was executed assumed only a single anomaly had occurred; because the sensors are in parallel, both would have had to fail in order to produce the observed data.

GeneSat-1

Launched in December 2006, GeneSat-1 is a technology demonstration spacecraft used to validate the use of research-quality instrumentation for *in situ* biological research and processing. Development and execution of the GeneSat-1 mission was led by NASA Ames Research Center, and NASA biologists served as the Principle Investigators for the associated science

experiment involving the evaluation of E. coli. metabolism in the microgravity environment.

NASA included significant university involvement in the GeneSat-1 mission. Stanford University graduate students performed early prototyping of the GeneSat-1 bus, Cal Poly students developed a P-POD launch ejector for the "triple-CubeSat" vehicle, and Santa Clara University students developed the entire ground segment and performed all on-orbit mission operations. In February 2007 with all primary mission objectives successfully complete, NASA turned full operational responsibility for GeneSat-1 over to Santa Clara University for the purposes of student training and engineering research until the satellite de-orbits in late 2007.

As shown in Figure 5, the GeneSat-1 space system includes the satellite itself, a primary communication station that uses an 18-meter parabolic dish, and Santa Clara's distributed internet-based command and control network. For primary science operations, the NASA Ames Multi-Mission Operations Center (MMOC) was configured to serve as a Control Node within this network.

With respect to MBR anomaly management, the demonstrations performed for GeneSat-1 built upon the Sapphire contributions and other work³⁶ in order to integrate the reasoning software with Santa Clara's operational command and telemetry analysis environment. A graphical environment, shown in Figure 6, was also developed in order to support system modeling as well as to convey reasoning outputs of the MBR agent; this was specifically done to provide decision support to a human operator³⁷. Extensions were also made to enhance to dynamic manner in which command and telemetry operations were simulated.

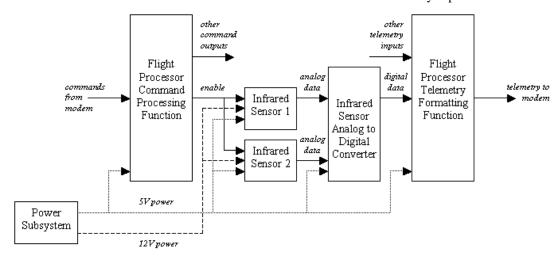


Figure 4: The Sapphire Infrared Sensor Configuration.



Figure 5: Elements of the GeneSat-1 Space Systems.

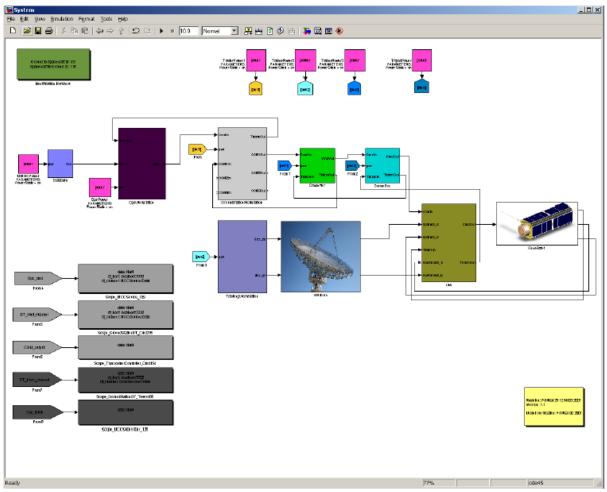


Figure 6: A View of the Graphical Model of the GeneSat-1 Space Systems.

Operationally, the refined reasoning system has successfully detected and diagnosed several anomalies. For example, several Internet link outages occurred (some during critical phases of the mission) between the communication station and the command and telemetry database located in the Ames MMOC. These were routinely detected, diagnosed, and resolved

through the instantiation of a local database at the communication station (appropriate early in the mission when the Control Node was located at the station).

In another compelling case, the system detected (in post-processing) a condition that had been missed by expert operators. This was an incremented vehicle command count indicating the reception of commands

for which the telemetry response was not received. Although the operational implications of this anomaly were benign, catching and subsequently tracking this deviation became instrumental in studying the link performance of the vehicle.

Algorithm Performance

Speed of processing is of obvious importance given the need to quickly return a space system to a healthy state in the event of an anomaly. Executing on a Pentium IV PC, the implementations of the algorithms used for the experiments presented here execute detection, diagnosis and resolution (for a specific diagnosis) on the order of seconds to tens of minutes. Broken down, detection generally occurs on the order of hundredths of seconds, diagnosis typically takes seconds or minutes, and resolution ranges from seconds to tens of minutes. This is for systems with ~30 components (which is the general resolution of operational anomaly management performed by humans) which typically require hundreds or thousands of mathematical constraints.

To put this performance level into the proper perspective, it is important to note two facts. First, state-of-the-practice anomaly diagnosis and resolution in the space industry often takes hours, days, or even weeks given the reliance on manual and experiential processing and the fact that the engineers that perform these tasks are generally not the operators that perform realtime command and telemetry operations. Second, our current implementations can be significantly improved given that computational performance has been sacrificed in order to promote exploratory implementation, to examine alternate modeling and computational approaches, and to assess the computational similarities in computing diagnoses and resolutions for different classes of anomalies. Furthermore, the current software runs as interpreted Matlab scripts and graphical Simulink models on multitasking computers rather than as a set of compiled executable functions on a dedicated workstation; optimization of the algorithms is expected to improve their computational speed by more than a factor of ten³⁸. Given these facts, we believe that MBR anomaly management is capable of providing valuable decision support in current space system operations environments.

Lessons Learned

These experiments yielded a number of "lessons learned" regarding the value of the model-based anomaly management strategy^{30,37,38}:

- MBR provided a systematic approach to evaluating the solution space for each anomaly management task.
- MBR served as a strong complement to the advantages and disadvantages of the experiential techniques routinely used by human operators, thereby making its use as a decision support tool particularly compelling. From this perspective, we see MBR not as a panacea for the anomaly management process but as a technology to complement other approaches that draw from experience and employ time-saving heuristics.
- Human interaction with the model-based reasoning system "rubbed-off" on the human operators and improved their ability to systematically evaluate anomaly scenarios.
- Discrepancies between the conclusions drawn by human operators using experiential techniques and those by the automated model-based system often took hours to resolve (e.g., for the human operators to understand why their conclusions were incorrect and/or incomplete). It is clear that the future acceptance of MBR systems (and, in fact, any high performance reasoning system) will require tools and mechanisms for providing insight into how the results were generated. This lesson is the motivation for current work in the graphical analysis environment.
- Development of the design models themselves was also tedious, again leading to our current work on our graphical modeling and analysis environment.
- The experiments demonstrated the power of using symbolic representations of engineering functionality in order to draw strong conclusions regarding the state of a system. There is a healthy tension between the need to use simple and abstract models to promote speedy computation and the need for modeling detail in order to provide the required level of reasoning resolution.

SUMMARY AND CONCLUSIONS

Our developed MBR techniques use fundamental knowledge of a system in order to compute reasoning conjectures relating to the existence of symptoms, diagnosis estimates, and resolution control actions. In doing so, these techniques provide a systematic and efficient framework for automated reasoning, which in turn can dramatically accelerate the analysis of anomalies with significantly improved results. Our work includes an expanded theoretical framework to address and conceptually unify the practical anomalies that are routinely experienced in the operation of space systems. This theory has been implemented through algorithms that currently execute as MATLAB

programs and which have been used as the engine for a Simulink-based graphical model of the system. Experimental evaluation of these systems has been used to verify functionality and to validate the benefits of the technology relevant to the speed and accuracy of reasoning conjectures. This work has been performed relating to the configuration control of the Sapphire and GeneSat-1 small spacecraft.

ACKNOWLEDGEMENTS

The authors thank the all of the university students, faculty and staff that have been an integral part of the development of the Sapphire and GeneSat-1 satellites and ground segments, with special acknowledgement to Ignacio Mas, Dr. Michael Swartwout, Tom Van Buskirk, and Daniel Schuet. We are also grateful to the Mr. John Hines, Ms. Karolyn Ronzano, and all of our partners on the NASA Ames Research Center GeneSat-1 mission team for their support in allowing us to conduct a portion of this work through the use of the GeneSat-1 spacecraft. Elements of this work have been supported through a wide variety of funding mechanisms to include the DoD NDSEG fellowship program, the National Science Foundation under Grant Numbers EIA0079815 and EIA00882041, the NASA Ames Research Center through Cooperative Agreement Numbers NNA04CK35A and NNA06CB13A, and through Santa Clara University's Technology Grants; any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DoD, the National Science Foundation, NASA, or Santa Clara University.

REFERENCES

- Hovanessian, S., Raghavan, S. and D. Taggart, "LIFELINE: A Concept for Automated Satellite Supervision," The Aerospace Corporation Report No. TOR-93(3516)-1, El Segundo, CA, 1993.
- G. Brown, D. Bernard, and R. Rasmussen, "Attitude and Articulation Control for the Cassini Spacecraft: A Fault Tolerance Overview," in Proceedings of the 14th AIAA/IEEE Digital Avionics Systems Conference, Cambridge, MA, November, 1995.
- 3. C. Kitts and M. Swartwout, "Experimental Initiatives in Space Systems Operations," in Proceedings of the Annual Satellite Command, Control and Network Management Conference, Reston, Virginia, 1997.
- 4. N. Ely and T. O'Brien, "Space Logistics and Reliability," Space Mission Analysis and Design, Eds. J. Wertz and W. Larson, London: Kluwer Academic Publishers, 1991, pp. 633-656.

- 5. State of the Space Industry, Space Publications, Reston, VA, 2005.
- Thurman, David A., David M. Brann, and Christine M. Mitchel, "An Architecture to Support Incremental Automation of Complex Systems," Proc 1997 IEEE Conference on Systems, Man and Cybernetics, Orlando, FL, 12-15 October 1997, pp. 1174-1179.
- 7. Leveson, Nancy G., and Everett Palmer, "Designing Automation to Reduce Operator Errors," Proc 1997 IEEE Conference on Systems, Man and Cybernetics, Orlando, FL, 12-15 October 1997, pp. 1144-1150.
- 8. Swartwout, Michael A., Kitts, Christopher A., and Rajesh K. Batra, "Persistence-Based Production Rules for On-Board Satellite Automation," IEEE Aerospace Applications Conference Proceedings, v 1, 1999, p 273-281.
- 9. Johnson, Steven, "Fault Protection Design for Unmanned Interplanetary Spacecraft," Guidance & Control 1987: Advances in the Astronautical Sciences, v63, pp. 3-18.
- Malina, R.F., "The EUVE Testbed: Innovations Toward Low Cost Science and Mission Operations Automation," Proc if the 47th International Astronautical Federation, Session IAA, 11.2: Autonomous Control for Small Satellites, Beijing, China, October 1996.
- 11. Stroozas, B.A.; Gunter, M.R.; Kaplan, G.C.; Nevitt, R.; Hartnett, K.; Malina, R.F. "EUVE Extends Automated 'Lights Out' Payload Operations to Spacecraft Platform," Systems, Man, and Cybernetics, 1997.
- 12. C. Kitts and M. Swartwout, "Beacon Monitoring: Reducing the Cost of Nominal Spacecraft Operations," Journal of Reducing Space Mission Cost, vol 1, no. 4, pp. 305-338, 2002.
- 13. C. Kitts, "A First Principles Approach for Managing Anomalies in Space Systems," IEEE Robotics & Automation Magazine, Sp. Issue on Automation Science, v 13 no 4, December 2006.
- 14. R. Reiter, "Theory of Diagnosis from First Principles," Artificial Intelligence, vol 32, no. 1 pp. 57-95, 1987.
- 15. J. de Kleer, A. Mackworth, and R. Reiter, "Characterizing diagnoses and systems," Artificial Intelligence, vol 56, pp. 197-222, 1992.
- 16. M. Genesereth, "The use of design descriptions in automated diagnosis," Artificial Intelligence, vol 24, pp. 411-436, 1984.
- 17. J. de Kleer and B. Williams, "Diagnosing multiple faults," Artificial Intelligence, vol 32, pp. 97-130, 1987.
- R. Doyle, S. Chien, U. Fayyad, and E. Wyatt, "Focused Real-time Systems Monitoring Based on Multiple Anomaly Methods," Proc 7th International

- Workshop on Qualitative Reasoning, Seattle, WA, May, 1993.
- D. DeCoste, "Automated learning and monitoring of limit functions," in Proceedings of the Fourth International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-97), Japan, July, 1997.
- L. Fesq, A. Stephan, and L. McNamee, "Modeling Power Systems for Diagnosis: How good is good enough," Proceedings 27th Intersociety Energy Conversion Engineering Conference, San Diego, CA, 1992, pp 203-208.
- B. Williams, and P. Nayak, "A Model-based Approach to Reactive Self-Configuring Systems," Proceedings AAAI National Conference on Artificial Intelligence, Portland, OR, 1996, pp. 971-978.
- G. Friedrich, G. Gottlob, and W. Nejdl, "Physical impossibility instead of fault models," Proceedings of the 1990 AAAI National Conference on Artificial Intelligence, Boston, MA, July, 1990, pp. 331-336.
- 23. P. Struss, and O. Dressler, "Physical Negation: integrating fault models into the general diagnostic engine," Proceedings of the International Joint Conference on Artificial Intelligence, San Mateo, CA: Morgan Kaufmann, 1989, pp. 1318-1323.
- 24. R. Davis, "Diagnosis based on description of structure and function," Proceedings of the 1982 AAAI National Conference on Artificial Intelligence, Pittsburg, PA, August, 1982, pp. 137-142.
- 25. M. Genesereth, "Diagnosis using hierarchical design models," Proceedings of the AAAI National Conference on Artificial Intelligence, Lost Altos, CA: Morgan Kaufmann, 1982, pp. 278-283.
- 26. J. Doyle, "A truth maintenance system," Artificial Intelligence, vol 12, pp. 231-272, 1979.
- 27. H. Ng, "Model-based, multiple fault diagnosis of time-varying, continuous physical devices," Proceedings of the 6th IEEE Conference on Artificial Intelligence Applications, Santa Barbara, CA, 1990, pp. 9-15.
- J. Crow, and J. Rushby, "Model-based reconfiguration: Toward an integration with diagnosis," Proceedings of the 1991 AAAI National Conference on Artificial Intelligence, San Mateo, CA: Morgan Kaufmann, 1991, pp. 836-841.
- 29. G. Friedrich, G. Gottlob, and W. Nejdl, "Formalizing the Repair Process," Proceedings of the European Conference on Artificial Intelligence, Vienna, August, 1992.
- 30. C. Kitts, "Theory and Experiments in Model-Based Space System Anomaly Management," Stanford University Ph.D. Dissertation, Palo Alto, CA, September 2005.

- 31. C. Kitts, "Model-Based Space System Anomaly Management Part 1: A First Principles Foundation," in draft for IEEE Transactions on Automation Science and Engineering.
- 32. C. Kitts, "Model-Based Space System Anomaly Management Part 2: Implementation and Application," in draft for IEEE Transactions on Automation Science and Engineering.
- 33. M. Swartwout, C. Kitts, R. Twiggs, B. Smith, T. Kenny, R. Lu, K. Stattenfield, R. Batra, and F. Pranajaya, "Sapphire: A Case Study in University Class Satellites," Accepted Acta Astronautica.
- 34. D. Schuet and C. Kitts, "A Distributed Satellite Operations Testbed for Anomaly Management Experimentation," Collection of Technical Papers AIAA 3rd "Unmanned-Unlimited" Technical Conference, Workshop, and Exhibit, Chicago, IL, September, 2004.
- 35. Kitts, C., Rasay, M., Mas, I., Mall, PJ., Van Buskirk, T., "Model-Based Anomaly Management for Spacecraft Mission Operations at the NASA Ames Space Technology Center." Proceedings of the 2nd International Conference on Space Mission Challenges for Information Technology, Pasadena CA, July 2006.
- Van Buskirk, T., and Weiler, K., Enterprise Class Mission Control Software Suite for the NASA Genesat-1 Spacecraft, Advisor: C. Kitts, Santa Clara University Undergraduate Thesis, June 2005.
- 37. Rasay, R., A Graphical Model-Based Reasoning Analysis Environment for Space System Anomaly Management, Advisor: C. Kitts, Santa Clara University Master's Thesis, June 2007.
- 38. E. Hall, "Maximizing Matlab Performance," University of Virginia Research Computing Support Center, Charlotteville, VA, 2003.
- 39. Schuet, D., A Distributed Satellite Operations Testbed for Anomaly Management Experimentation, Advisor: C. Kitts, Santa Clara University Master's Thesis, June 2004.