



Modular Architecture for Hybrid Diagnostic Reasoners

Han G. Park^{1,2}, Anthony Barrett¹, Erwin Baumann², Michael Grage², Sriram Narasimhan³
Jet Propulsion Laboratory¹, Northrop Grumman Corporation², NASA Ames Research Center³
Han.Park@ngc.com



Abstract

This paper describes technical progress made in maturing ISHM technologies for hybrid diagnostic reasoners under NASA's Exploration Systems Research & Technology (ESR&T) program by Northrop Grumman Corporation and its NASA partners at ARC, GRC, JPL, and JSC. A layered architecture was developed that supports both distributed and hierarchical reasoning as well as interoperability and interchangeability of reasoning algorithms. A hybrid reasoner using this architecture was to be demonstrated performing automated fault detection and isolation in an "Apollo 13"-like failure scenario.

1. Introduction

Future space exploration systems, such as the Crew Exploration Vehicle (CEV), lunar habitats, and Mars transfer vehicles, will require very robust on-board diagnostic and prognostic reasoning capabilities to achieve the safety margins and self-reliance required for these increasingly ambitious missions. While many diagnostic and prognostic reasoners exist, only a few have been validated in contexts other than very constrained laboratory demonstrations. Furthermore, it is likely that an actual implementation of an Integrated Systems Health Management (ISHM) system will require that many different types of reasoners work together to generate candidate diagnosis for widely varying types of failures. In essence, a hybrid reasoning architecture, in which different reasoners can communicate with each other, is required. Such architectures have not been adequately demonstrated to date and require further development. Furthermore, most reasoners have been tested only in isolation with little regard for interoperability and interchangeability.

Faced with these issues, Northrop Grumman Corporation and its NASA partners (Ames Research Center - ARC, Glenn Research Center - GRC, and Jet Propulsion Laboratory - JPL, Johnson Space Center - JSC) led a project under NASA's Exploration Systems Mission Directorate (ESMD) Exploration Systems

Research and Technology (ESR&T) program to mature and validate critical ISHM technologies for application to future space exploration systems. A key technical objective of this effort was to develop and demonstrate advanced diagnostic and prognostic hybrid reasoning algorithms produced by combining existing nonproprietary algorithms and software. This included developing architectures and interfaces required to integrate different types of reasoning algorithms and achieve a level of "plug-and-play" interoperability and interchangeability.

Since no single reasoning technology currently available has been shown to effectively handle all types of data (e.g. quantitative, qualitative, static, dynamic) and isolate complex interacting failures while operating in real-time on a distributed network, the thesis of this project is that hybrid reasoning systems offer a practical way of achieving these goals and providing the core of robust and flexible ISHM systems. Hybrid reasoning systems, if properly designed, should exhibit synergies that leverage the strengths and compensate for any weaknesses in the component diagnostic algorithms. For instance, a Kalman filter might be used to monitor the low-level closed-loop controlling the temperature of a heater (continuous variable state) while a model-based reasoner (MBR) [1] is used to diagnose faults associated with the heater switches (discrete states). Such continuous and discrete state hybrid reasoners have been demonstrated previously [2]. In addition, if hierarchical reasoning is applied to system-level fault isolation by reasoning across multiple subsystem diagnoses, hybrid reasoning is practically unavoidable since the subsystems are likely to embody different types of reasoners. For example, an ISHM system for a future manned exploration vehicle might have the power subsystem monitored by an MBR system, the Environmental Control & Life Support (ECLS) subsystem by a neural-network, and system-level reasoning and crew interface supported by a rule-based system.

This paper describes technical progress in the areas of architecture, interfaces, hybrid reasoner

integration, and planned demonstrations. Due to NASA budget constraints and priorities, the demonstration of the hybrid reasoner was not completed. However, significant background work was conducted in developing a general architecture for hybrid reasoner integration as well as the design of a proposed hybrid reasoner planned for demonstration on a simulation of an “Apollo-13”-inspired multi-level cascading failure scenario.

2. Architectures for health management

Integrating various types of diagnostic / prognostic algorithms to create a hybrid reasoning system requires an architecture that can satisfy competing demands. The architecture must be general enough to support a wide range of algorithms and diagnostic approaches, yet specific enough to maintain consistent states and mapping between different classes of reasoners in order to support interoperability and interchangeability. It must also be capable of supporting both distributed and hierarchical reasoning.

Two existing open standards which can provide the foundation for hybrid health management reasoning system architectures are: International Organization for Standardization (ISO) 13374 [3] and Machinery Information Management Open Systems Alliance (MIMOSA) Open System Architecture for Condition Based Maintenance (OSA-CBM) [4][5].

The ISO 13374 standard addresses condition monitoring and diagnostics of machines. The intent of ISO 13374 is to provide the basic requirements for open software specifications which will allow machine condition monitoring data and information to be processed. This standard establishes general guidelines for software specifications related to data processing, communication, and presentation of machine condition monitoring and diagnostic information. ISO 13374 defines machine condition assessment using six distinct, layered, processing blocks. OSA-CBM is an implementation of ISO 13374 targeted for real-time, on-board, and embedded condition-based maintenance system architectures, including those involving logically and/or physically distributed CBM software components.

These standards embody a layered architecture where layers correspond to levels of signal, data, or information processing and information propagates between layers in a defined manner. The layers are:

Layer 1, Data Acquisition - Represents the software module that provides system access to digitized sensor or transducer data.

Layer 2, Data Manipulation - Performs single and/or multi-channel signal transformations along with specialized feature extraction algorithms.

Layer 3, State Detection - Identifies components that are outside anticipated operational envelopes and flags these components for assessment.

Layer 4, Health Assessment - Provides a health assessment of the subsystem (e.g., nominal or off-nominal/degraded in some specified way).

Layer 5, Prognostics Assessment - Projects the current health state into the future including estimates of remaining useful life.

Layer 6, Advisory Generation – Provides appropriate advisory, caution, and warning messages in response to outputs from the lower level Fault Detection, Isolation, and Recovery (FDIR) functions, and handles requests from external control to respond to the current health situation.

These standards, particularly OSA-CBM, were to be used as the basis for the hybrid system architecture. However, the original intent of OSA-CBM was to serve as an architecture for aircraft maintenance and safety critical health monitoring. Thus the layers correlate most closely to maintenance functions rather than critical diagnostic reasoning functions. This presented several problems. First, the definitions of layers 2, 3, and 4 created some confusion as to which layer a reasoner belongs. For example, BEAM, which is a signal-level analysis system, might be characterized as belonging to either layer 2 or 3. Furthermore, the layer definitions were too coarse, especially between layers 3 and 4, to represent the typical reasoning steps for model-based reasoners. Finally and most importantly, the existing layers reflected no obvious concept of hierarchical reasoning. It was therefore initially unclear how to correlate hierarchical reasoner operation to this layered representation.

3. Architecture for hybrid reasoning

To satisfy the demands of hierarchical reasoning and facilitate modularity of reasoners such as a MBR system, we proposed the architecture depicted in Figure 1. The architecture is layered like OSA-CBM, but with the layers now paralleling the processing steps of consistency-based reasoning [6]. In consistency-based reasoning, the goal is to identify the faulty components that best account for the difference between predicted normal device behavior and observed (abnormal) behavior. Predicted behavior is inferred from a formal model of normal structure and behavior of the device. While the layers are based on consistency-based reasoning, they remain general

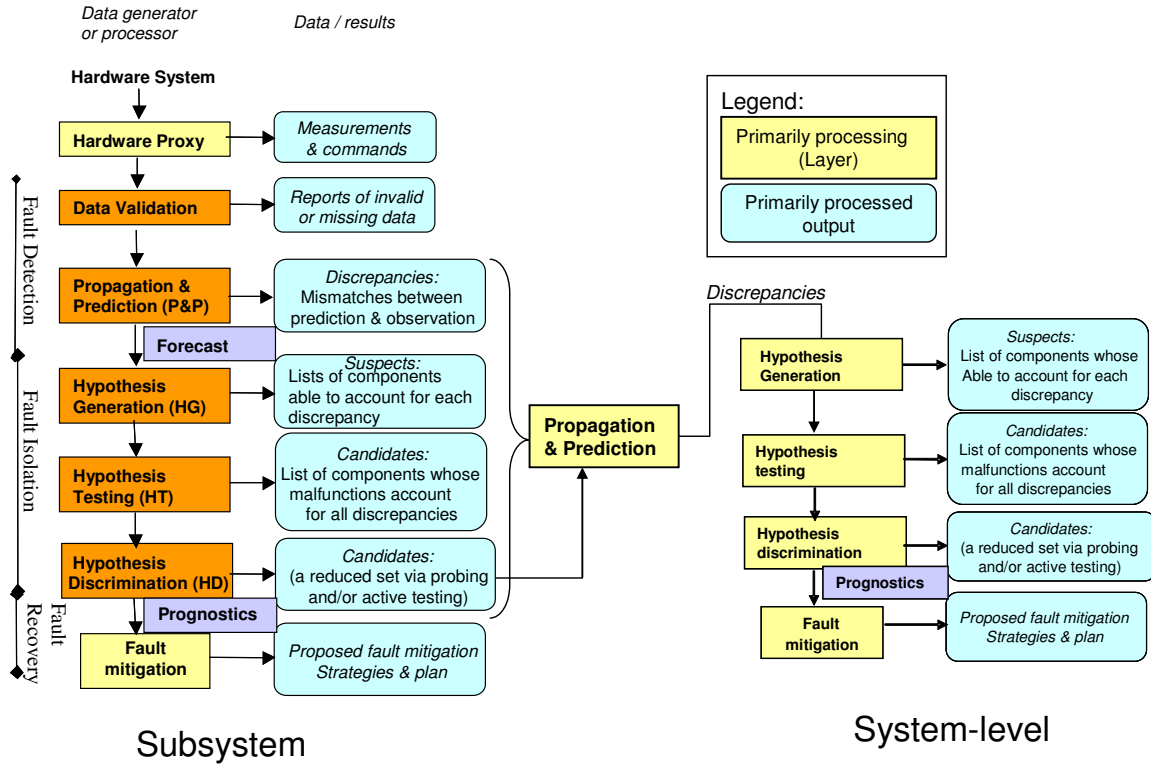


Figure 1. Proposed layered architecture for hybrid resoner.

enough to accommodate a large number of diagnostic reasoning paradigms and algorithms. The small changes in the layers and their definitions enable hierarchical reasoning while maintaining consistent states and mapping between layers.

The development of interfaces and standards centered on the following five main reasoning layers in the architecture as indicated in Figure 1 as dark (brown colored) boxes:

- Data Validation (DV)
- Propagation and Prediction (P&P)
- Hypothesis Generation (HG)
- Hypothesis Testing (HT)
- Hypothesis Discrimination (HD).

After receiving the sensor data, the DV layer determines whether the information is valid and what state they are in. This may be a simple binning operation where continuous values are translated into coarse discrete values, or a complex operation such as sensor state estimation using Kalman filters or feature-based algorithms. The P&P layer generates a list of discrepancies by comparing predicted sensor values to actual observed values. Traditionally, these two layers are described as fault detection. The HG layer

generates a suspect for each discrepancy, which is a list of components where one's failure accounts for the discrepancy. In consistency-based reasoning, this function may be called conflict generation. The HT layer generates a list of candidates, each of which is a minimally sized set of malfunctioning components that account for all observed discrepancies. In consistency-based reasoning, this function may be called diagnosis. Finally, the role of the HD layer is to refine the list of candidates by gathering more observations, i.e., evidence, either through passive or active means. The combination of layers HG, HT, and HD is traditionally referred to as fault isolation.

Consistent states and mapping between different classes of diagnostic algorithms are maintained by establishing standards for the processed output generated at each layer. For example, the P&P layer generates discrepancies that follow the standard, regardless of whether the P&P layer represents a rule-based, neural network, or Kalman filter algorithm. This enables interoperability and interchangeability of hybrid resoner components. Furthermore, this architecture supports both distributed and hierarchical reasoning. It also supports distributed reasoning by allowing the layers to operate in parallel. For example, it is possible for multiple independent algorithms to generate discrepancies in the P&P layer. However, the

powerful feature of this architecture is that it naturally supports hierarchical reasoning which enables system-level health monitoring and isolation of faults across subsystems. At the subsystem-level, for example, the output of the HT or HD layer is a list of candidates (i.e., diagnoses for the subsystem). At the system-level these sets of subsystem candidate diagnoses can be used as the set of discrepancies, i.e., the inputs for the HG layer. In effect, the entire subsystem-level reasoner appears as a P&P layer to the system-level reasoner as shown in Figure 1. This characteristic of the architecture enables the creation of hierarchies to any arbitrary number of levels, extending from simple subsystem- to-system hierarchies through arbitrarily extensive and complex “system-of-systems” hierarchies.

3.1. Comparison to OSA-CBM

The proposed reasoning layers are mapped to the OSA-CBM layers as shown in Figure 2. The lowest layer, Hardware Proxy, spans two OSA-CBM layers, Data Acquisition and Data Manipulation. The goal of the Hardware Proxy layer is to acquire signals from hardware and convert them into software readable format. The two layers, DV and P&P, are spanned by a single OSA-CBM layer, State Detection. The essential functions of these layers are to check for consistency of the sensor data and identify mismatches between the predicted and observed data.

Significant differences in the arrangement of the layers are observed in the HG, HT, and HD layers which are simultaneously covered by the Health and Prognostic Assessment layers in OSA-CBM. In the OSA-CBM architecture, health assessment and prognostic assessment are represented as serial operations. In the proposed hybrid architecture, both are covered by the HG, HT, and HD layers because the reasoning process is essentially identical for diagnosis and prognosis. The only difference is that, for prognostics, the reasoning is carried out using forecast sensor data rather than current data. As previously noted, splitting the Health Assessment layer into HG, HT, and HD layers allows more modularity in the algorithms and natural hierarchies to be created. The processed outputs of HT or HD from a lower level of the hierarchy can be thought of as the P&P layer for the hierarchy one level above.

Finally, the Fault Mitigation layer performs one of the roles of the OSA-CBM Advisory Generation layer. In OSA-CBM, the advisory layer: 1) Informs the human and/or machine of faults. 2) Provides advice on how to handle the current fault situation. The Fault Mitigation layer essentially performs the second of

these functions. In the proposed hybrid architecture, there is no explicit layer dedicated to generating output for humans and machines. This function is expected to be carried out by external processes that intercept messages flowing between the layers.

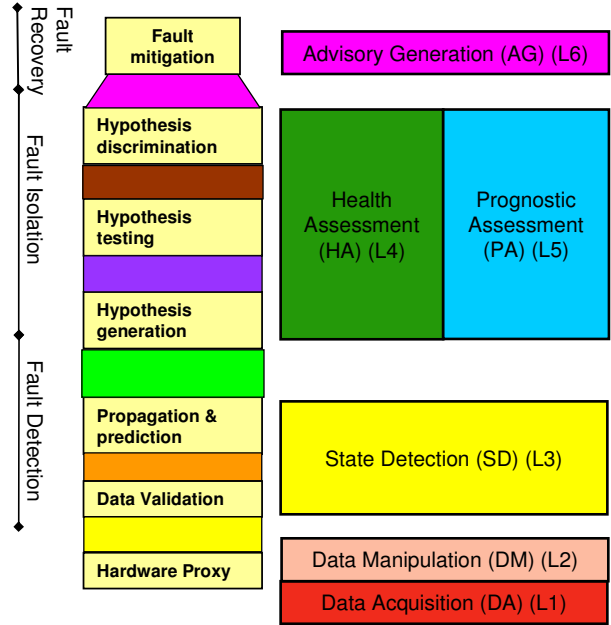


Figure 2. Reasoner layers comparison to OSA-CBM

The benefits of the layer definitions proposed here are that they describe more accurately, and in greater detail, the reasoning process required to generate a diagnosis or health assessment. In a sense, this formulation has exposed the sub-layers of the Health Assessment layer of OSA-CBM. Exposing these layers facilitates the design of modular reasoning algorithms without having to build a monolithic, large, and/or complex software system to generate a complete health assessment. For instance, if a faster and more efficient conflict generation algorithm is developed, it can be readily inserted into the HG layer without requiring changes to any of the other layers. Finally, the proposed layered architecture now more explicitly embodies the concept of a hierarchy. Within the standard OSA-CBM architecture it might be possible to construct a hierarchy of sorts by carefully crafting the output messages of the Advisory Generation layer for transmission to the Health Assessment layer of the next higher level in the hierarchy. However, this is a relatively cumbersome ad-hoc method that may also violate basic OSA-CBM concepts.

4. Demonstration of hybrid reasoning

The proposed architecture and hybrid reasoner were to be demonstrated on an “Apollo-13”-like scenario. The failure mode of the Apollo 13 lunar mission was an actual multi-point, cascading failure that is representative of an important class of complex, unexpected, life threatening, and failure modes. These complex failure modes are still survivable given sufficiently rapid diagnosis, fault isolation, and selection and execution of recovery options as proven by the cooperative efforts of the Apollo 13 mission control team and crew [7][8]. However, if the communication system had also failed, or the crew was on a mission to Mars, for example, with communication delay times on the order of tens of minutes, the outcome is likely to have been less desirable. Such events argue for an increased level and sophistication of automated on-board fault detection, isolation, and recovery for increasingly ambitious missions, such as those planned for the Moon and Mars. Hybrid diagnostic reasoners can provide a key part of the solution to this problem.

Because the failure is well documented, the scenario can also serve as a reference point by which to compare the manual detection and diagnosis times from Apollo 13 with the performance of a hybrid-reasoner-based ISHM system. To demonstrate the capabilities of a hybrid diagnostic reasoning system in this context, the Environmental Control and Life Support (ECLS), electrical power distribution, and fuel-cell cryogenic-feed subsystems were chosen as the focal points for the demonstration as they represented the critical subsystems involved in the Apollo 13 failure.

4.1. Reasoning technologies

The constituent diagnostic algorithms provide the foundation of any reasoner technology. The selection of candidate algorithms was based on three factors: 1) Diversity in the types or classes of the reasoning algorithms and their ability to analyze a wide variety of fault symptoms; 2) Maturity level of the software; 3) Existing algorithm expertise among the project team members. The selected algorithms were:

MEXEC (Model Executive) – A model-based diagnostic engine that uses knowledge compilation techniques for analyzing a set of N observations to rapidly determine the most likely states of a system in real-time. A system is modeled as a network of interacting components, each of which is modeled as a state machine. MEXEC can deal with common issues including limited observability and time-delayed symptoms. MEXEC's diagnostic precision improves

with increasing N at the cost of increased computational overhead [9].

BEAM (Beacon-based Exception Analysis for Multimissions) - A feature-based / statistical anomaly detector for real-time fault detection and characterization. An allowed set of behaviors is “learned” and deviations from this are noted and examined. For each phase of normal operation, BEAM is trained on the nominal subsystem data. Any deviation from the nominal behavior is tagged as anomalous. A simple reasoner then generates a list of candidates (i.e., components or signals that are responsible for the anomaly) and tries to identify the type of anomaly. In essence, BEAM takes quantitative measurements and converts them into qualitative measurements [10][11].

SHINE (Spacecraft Health Inference Engine) – A rule-based system designed to meet the needs of a reusable inference engine (expert system) for the purpose of monitoring, analysis, and diagnosis of real-time and non-real-time systems. The knowledge base (set of rules) is encoded as a set of propositional logic statements. SHINE was designed to be efficient enough to operate in a real-time environment.

HyDE (Hybrid Diagnostic Engine) – A model-based diagnostic engine that uses candidate generation and consistency checking to diagnose discrete faults in stochastic hybrid systems. HyDE can also deal with common issues such as limited observability, sensor noise, and time-delayed symptoms [12].

The reasoners encompassed three classes of algorithms: 1) feature-based reasoning; 2) rule-based reasoning; and 3) model-based reasoning. These choices were deemed sufficient to demonstrate and validate the architecture and interfaces.

4.2. Integration

As mentioned previously, the selection of reasoners was based upon their availability and maturity, the expertise of the project team members, and the classes of reasoners represented. The strengths and weaknesses of each reasoner were compared to the characteristics and response times associated with subsystem and system-level diagnostic requirements. This evaluation led to the assignment of reasoners to subsystems and/or system-level diagnostic tasks that would take best advantage of those capabilities. For example, the SHINE-BEAM-MEXEC combination provided the fastest response time and was therefore chosen to monitor the power distribution system because it had the most demanding response time requirements (on the order of tens of milliseconds).

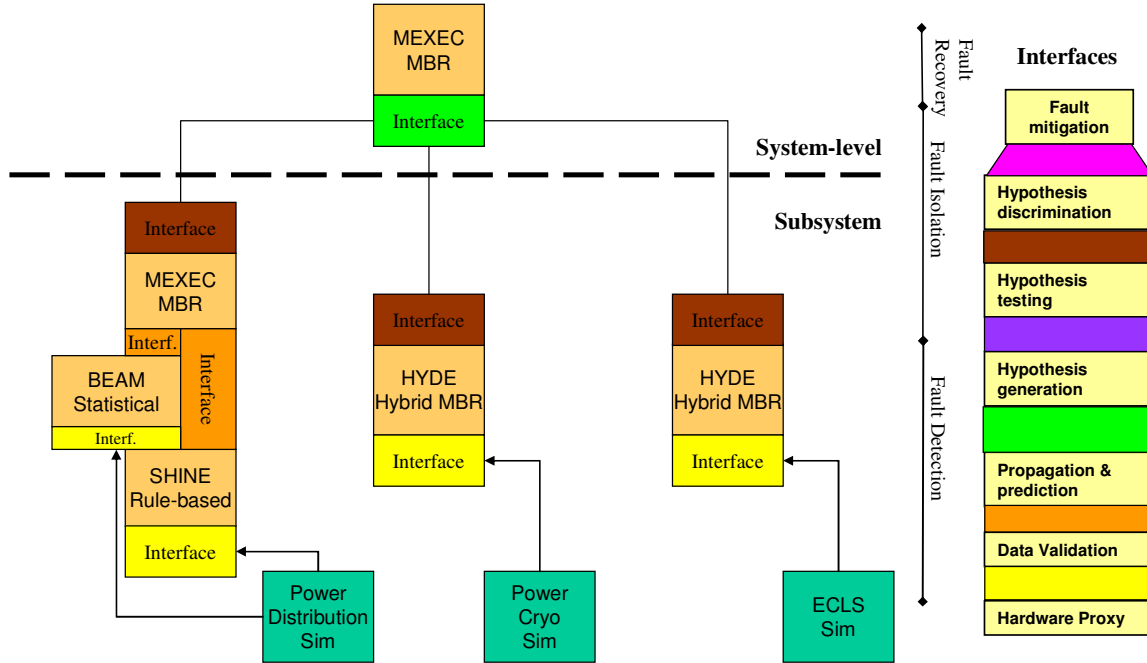


Figure 3. Demonstration reasoner architecture and interfaces.

The resulting hierarchical architecture, mapping of the reasoners to subsystems for the demonstration, and interfaces for each reasoner are shown in Figure 3.

The HyDE reasoner was selected to monitor the power cryogenic and the environmental control and life support (ECLS) subsystems. The HyDE reasoner spans all of the reasoning layers, from DV to HT. HyDE was originally based on its own custom internal architecture. The effort to adapt HyDE to the proposed hybrid architecture was delayed due to resource limitations. Therefore, HyDE was used primarily to demonstrate interface standards and the hierarchical reasoning capabilities facilitated by the hybrid architecture.

The first set of reasoners to demonstrate the layered reasoning approach is the SHINE-BEAM-MEXEC combination. SHINE encompassed the DV layer, where it translates the continuous sensor readings into discrete states and report the readings to MEXEC. Also residing in the DV layer, BEAM would detect any anomalies in the dynamics of the associated sensor(s), which would be passed on as sensor state measurements to MEXEC in the P&P layer. MEXEC, which also includes the HG and HT layers, would employ the sensor state information from the DV layer (BEAM and SHINE) to generate candidate diagnoses. The layers supported by the reasoners are listed in Table 1.

The candidate diagnoses generated by the subsystem reasoners, HyDE and SHINE-BEAM-MEXEC, would be sent as states of an abstract subsystem sensor the MEXEC system-level diagnostic reasoner. The role of the system-level MEXEC reasoner is to narrow down the list of subsystem candidates by eliminating those that are not consistent with all of the diagnoses of other subsystems. MEXEC was chosen for the system-level reasoner because model-based reasoners provide greater formalism for modeling the subsystem interactions. System-level reasoners have traditionally been developed using rule-based systems which are typically very system specific, sometimes ad-hoc, and not easily generalized. One of the related research goals included the development of a general framework for applying model-based reasoners to system-level reasoning tasks.

Table 1: Reasoner and layer correspondence

Layer	HyDE	SHINE	BEAM	MEXEC
D V	X	X	X	
P & P	X			X
H G	X			X
H T	X			X

4.3. System-level reasoning

An area that received special attention was the information exchanged between subsystem and system-level reasoners. Typically, the output of the HT layer at the subsystem level is a set of candidate diagnoses. A problem for the system-level reasoner is the sheer number and combinations of candidate diagnoses that could be received from subsystems. The proposed solution to this problem is to reduce the number of candidate diagnoses conveyed to the system-level reasoner by using abstract states to instead describe the categories in which the set of subsystem diagnoses fall. That is, instead of reporting every set of candidate diagnoses, the subsystem reasoners would report their diagnoses in the form of abstract candidate diagnoses “states”. The system level reasoner then finds the consistent states for the subsystems and these found states subsequently narrow the number of subsystem diagnoses. For instance, suppose that subsystem diagnoses for power distribution in Apollo 13 resulted in 10 candidates partitioned as in figure 4. While these candidates are consistent with all information that the subsystem diagnoser has access to, the system level has access to diagnoses of the power source. If the source cannot provide power, then the distribution system has no power either, resulting in only one candidate being globally consistent.

The main motivation behind the abstract subsystem states is to keep the system level from having to reason with too many low level details. If the subsystem reasoner does not provide an abstracted description of its diagnosis, and could instead identify a potentially huge set of candidate diagnoses, the system-level reasoner would effectively need to “understand” and internally track every potential state of the subsystems to perform its own diagnosis. This, in effect, would also mean that the system-level reasoner would require a redundant model of the entire system, eliminating the need for separate subsystem reasoners and negating the advantages of hierarchical reasoning in reducing computational and modeling costs.

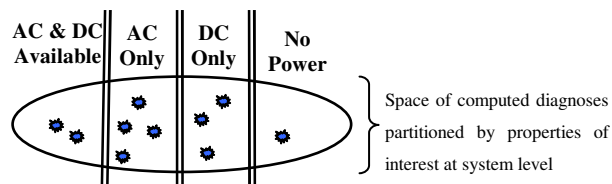


Figure 4. Power distribution subsystem state abstractions for system-level diagnostic evaluation.

4.4. Interface

The actual implementation of the hybrid reasoners requires a definition of the interface standards (messages) between the reasoning layers. The main consideration for the initial draft was the essential information that needs to be exchanged between layers to maintain consistent states and mapping. The initial draft did not consider message types and any schema.

The essential information that needs to be passed from one layer to the next higher layer was identified as:

1. DV: time, sensor list, states/values, and confidence (likelihood)
2. P&P: time, sensor list, mismatched states/values, and confidence (likelihood)
3. HG: time, component list, suspected mode states, and confidence (likelihood)
4. HT: time, component list (or abstracted component list), mode states, confidence (likelihood)

Information exchange between the reasoning layers was to be implemented in a manner consistent with the proposed OSA-ISHM architecture. This would be accomplished by mapping the reasoning layers into corresponding OSA-ISHM layers as shown in Figure 2. For example, the P&P reasoning layers map into a single OSA-ISHM State Detection (SD) layer while the HG and HT layers map into the OSA-ISHM Health Assessment (HA) layer. SD and HA messages would then be defined and used to convey state abstractions and diagnostic information between layers.

5. Conclusion

The essential technologies and architecture for creating hybrid reasoners using nonproprietary algorithms and software were presented. The architecture is layered in a manner inspired by, and similar to, OSA-CBM, but with the layers paralleling the reasoning steps of consistency-based reasoning methods. While the layers are based on consistency-based reasoning, they remain generic enough to accommodate a large number of diagnostic reasoning paradigms and algorithms. However, the layers are also specific enough to maintain consistent states and mappings between different classes of reasoners to support interoperability and interchangeability. The proposed layers are also capable of supporting both distributed and hierarchical reasoning architectures.

The layered architecture as well as the hybrid reasoner was to be demonstrated on an "Apollo 13"-like multi-subsystem cascading failure scenario. A MEXEC-BEAM-SHINE combination hybrid reasoner was being developed to monitor the power distribution subsystem, while the HyDE hybrid reasoner was to monitor the ECLS and power cryogenic subsystems. A second MEXEC process was to serve as the system-level reasoner. Although the hybrid reasoning system was not fully implemented due to the premature cancellation of the project, significant technical progress was made by the project team in the areas of architecture, interfaces, hybrid reasoner integration, demonstration concept development, and subsystem simulations.

6. Acknowledgements

This project was funded under the NASA's Exploration Systems Mission Directorate (ESMD) Exploration Systems Research and Technology (ESR&T) program. The authors would like to thank Dr. Robert Morris of NASA Ames Research Center for his guidance and support as NASA's Contracting Office Technical Representative (COTR), and Dan Dvorak of JPL for his original discussions on diagnostic reasoning methods.

7. References

- [1] Williams, B.C., Nayak, P.P., "A Model-based Approach to Reactive Self-Configuring Systems," *Proceedings of the National Conference on Artificial Intelligence*, 1996.
- [2] Park, H.G., Cannon, H., Bajwa, A., Mackey, R., James, M., Maul, W., "Hybrid Diagnostic System: Beacon-based Exception Analysis for Multimissions - Livingstone Integration," *Machinery Failure Prevention Technology Conference*, Virginia Beach, Virginia, April 2004.
- [3] International Standard ISO 13374-1: Condition monitoring and diagnostics (CM&D) of machines – Data processing, communication and presentation – Part 1: General Guidelines. <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=21832>
- [4] MIMOSA References: <http://www.mimosa.org/>
- [5] OSA-CBM References: <http://www.osacbm.org/>
- [6] Reiter, R., "A Theory of Diagnosis from First Principles," *Artificial Intelligence*, vol. 32, no. 1, pp. 57-96, 1987.
- [7] Apollo Operations Handbook Block II Spacecraft, Systems Data, Section 2, Subsection 2.6 Electrical Power System, October 15, 1969.
- [8] Cortright, Edgar M., "Apollo 13 Review Board Final Report," June 15, 1970.
- [9] Barrett, A., "Model Compilation for Real-Time Planning and Diagnosis with Feedback," *International Joint Conference on Artificial Intelligence (IJCAI 2005)*, Edinburgh, Scotland. July 2005.
- [10] Park, H.G., Mackey, R., James, M., Zak, M., Baroth, E., "BEAM: Technology for Autonomous Vehicle Health Monitoring," *CS/APS/PSHS/MSS JANNAF Meeting*, Destin, Florida, April 2002.
- [11] Park, H.G., Zak, M., "Grey-box Approach for Fault Detection of Dynamical Systems," *ASME Journal of Dyn. Sys., Meas., & Control*, Vol. 125, pp. 451-454, 2003.
- [12] Narasimhan, S., Dearden, R., Benazera, E., "Combining Particle Filters and Consistency-based Approaches for Monitoring and Diagnosis of Stochastic Hybrid Systems," *15th International Workshop on Principles of Diagnosis (DX04)*, Carcassonne, France, June 2004.