Functional Fault Modeling of a Cryogenic System for Real-Time Fault Detection and Isolation

Bob Ferrell,[1] Mark Lewis[2] and Jose Perotti[3]
NASA, Kennedy Space Center, FL, 32899

Rebecca Oostdyk[4]
ASRC Aerospace, Kennedy Space Center, FL, 32899

and

Barbara Brown[5]
NASA Ames Research Center, Kennedy Space Center, FL,32899

The purpose of this paper is to present the model development process used to create a Functional Fault Model (FFM) of a liquid hydrogen (LH2) system that will be used for real-time fault isolation in a Fault Detection, Isolation and Recover (FDIR) system. The paper explains the steps in the model development process and the data products required at each step, including examples of how the steps were performed for the LH2 system. It also shows the relationship between the FDIR requirements and steps in the model development process. The paper concludes with a description of a demonstration of the LH2 model developed using the process and future steps for integrating the model in a live operational environment.

I. Introduction

WHEN setting out to model and/or simulate a complex mechanical or electrical system, a modeler is faced with a vast array of tools, software, equations, algorithms and techniques that may individually or in concert aid in the development of the model. Mature requirements and a well understood purpose for the model may considerably shrink the field of possible tools and algorithms that will suit the modeling solution. Is the model intended to be used in an offline fashion or in real-time? On what platform does it need to execute? How long will the model be allowed to run before it outputs the desired parameters? What resolution is desired? Do the parameters need to be qualitative or quantitative? Is it more important to capture the physics or the function of the system in the model? Does the model need to produce simulated data? All these questions and more will drive the selection of the appropriate tools and algorithms, but the modeler must be mindful of the final application throughout the modeling process to ensure the model meets its requirements without needless iterations of the design. The purpose of this paper is to describe the considerations and techniques used in the process of creating a functional fault model of a liquid hydrogen (LH2) system that will be used in a real-time environment to automatically detect and isolate failures.

The LH2 functional fault model was the first model developed by the Fault Detection, Isolation and Recovery (FDIR) project. FDIR is funded by NASA's Exploration Technology Development Program (ETDP), and its is purpose is to mature fault detection, fault isolation, anomaly detection, and prognostics technologies for use in the new Constellation Program and future extra-planetary missions. FDIR is intended and designed to be integrated with Ground Operations to automate fault detection and isolation during maintenance and checkout as well as

[1] Electronics Engineer, Advanced Systems Branch, Mailstop: NE-E9
[2] Electronics Engineer, Advanced Systems Branch, Mailstop: NE-E9
[3] Chief, Advanced Systems Branch, Mailstop: NE-E9
[4] Electrical Engineer, Advanced Electronics and Technology Development, Mailstop: ASRC-25
[5] NASA-Ames Resident Manager at KSC, Ames Resident Office, ARC

launch countdown activities of ground and launch vehicle systems[1]. The FDIR architecture supports the integration of several ISHM capabilities, but this paper will focus on the modeling for fault isolation.

II. Requirements

A functional fault model (FFM) is ultimately used to compute a list of suspect or bad items, called an ambiguity group, ranked by likelihood based on the results of diagnostic tests. However, the FDIR project and Ground Operations customer has levied specific requirements on the fault isolation capabilities of the LH2 FFM. These requirements can be broken down into two categories: model requirements and operational requirements. The model requirements include:

1) Model shall have clear mapping back to physical system to aid in initial model validation by system experts and maintainability and sustainability by system design engineers.
2) The model shall be capable of isolating to multiple levels of resolution for the vehicle and ground systems (i.e. failure mode, component, line replaceable unit, etc.).
3) Modeling techniques and practices shall be scalable for a large, integrated model that encompasses vehicle systems, ground systems, and facility infrastructure. The integrated model will have an estimated 40,000 failure modes and 50,000 test points if it includes the ground systems, launch vehicle and Orion capsule.

The operational requirements are intended for the real-time use of the model. They include:

1) The reasoner shall diagnose multiple independent faults that occur simultaneously[2].
2) The reasoner shall provide a minimal component set based on a particular fault[2].
3) The reasoner shall re-configure the failure effect propagation paths to reflect the current mode of operation and system configuration within one second of a mode change.
4) Detected systems faults shall be isolated to the level required for recovery of function.
5) Detected faults shall be isolated to the level required for removal of line replaceable unit (LRU).
6) Integrated system faults shall be diagnosed to the level required for removal of line replaceable unit (LRU).
7) Fault isolation results shall be provided within 1 second of fault detection. The results shall include lists of suspect or bad items from the model.

III. Choosing the Right Model for the Application

Automated fault isolation requires the operational behavior of the system and its components to be defined so that it can be compared to the real-time system operation. The operational behavior is best captured in a model, and the type of model that is selected will be dependent on the application, resources and computing platform for the ISHM system.

For the FDIR project, several model-based diagnostic approaches were considered before the functional fault model (FFM) was chosen to represent the ground and launch vehicle systems. A physics-based model by which live data can be compared to theoretical values is the most accurate approach to detecting failures of the system, but the complexity of the ground and vehicle systems in all of their potential configurations and mission phases, as well as their dynamic operations, would make the physics model difficult to validate by subject matter experts as specified by the first model requirement. If the modeled systems had archives of historical data, the physics model could be compared to past operational behavior in lieu of an expert review, but the Constellation ground and vehicle systems have yet to be built or operated. Therefore, the physics-based model was not chosen due to the lack of dynamic operating history and validation methods.

Another candidate model was a rule-based expert system. The intent of the models is to aid engineers and operators who are monitoring the systems by providing information about the health of the components and the entire system. An expert system would be able to determine the state of the components and system in a reliable, repeatable manner assuming that its knowledgebase was comparable to that of the engineer or operator. However, the process of translating the engineer's expertise into a model requires a significant amount of the engineer's time. Since the modeling effort was meant to be carried out by a group of non-subject matter experts, the rule-based expert system modeling approach was discarded.

The FFM was selected because it allowed the modeling team to review system design documentation independently from the operators and engineers, create a model that resembles the schematic diagrams that are familiar to the experts, and have the experts verify the model without a large time commitment. Functional fault modeling involves capturing failure modes that have been identified both at design time and operationally. The Failure Mode Effects Analysis (FMEA) provides design-time failure modes, and problem reporting and corrective action databases and manufacturer datasheets provide insight into operational failures that are likely to occur or have been encountered for similar components in system. Once the failure modes have been catalogued, they can be placed in a model that maps the effects of the failures on system operation. Essentially, a FFM is responsible for identifying the failure effect propagation paths (FEPPs) from a failure mode to the observation point where it is detected. The FFM is then used in real-time to infer which failure modes or failed components could cause the observed behavior of the system.

Although the FFM has many advantages, there are two weaknesses which must be documented. The first, which has not been addressed by the FDIR project, is that the FFM is only germane for fault isolation during the steady-state operation of the system it models. Transitions between the system's physical configurations or mission phases are discretized into unique set of FEPPs. Fault isolation cannot accurately deduce the cause of a detected fault while the physical system is reconfigured and the FEPPs are in flux. Another weakness of the modeling technique is that the FFM only captures known failures. In order to supplement the FFM, the FDIR architecture also includes a data-driven model that detects anomalies. The data-driven model was trained on data from similar ground and vehicle systems to create a knowledgebase of in-family behavior. After sufficient training on nominal data, the data-driven model provides a measure of how closely the data it is monitoring matches the training data. The data-driven model is not able to use anomalous scores to isolate to a suspect component or system, but it does provide information about which measurements are contributing the most to the anomalous scores so that an operator or engineer can be alerted to a potential problem. The data-driven and functional fault models are complementary technologies that cover both known and unknown conditions of the system.

IV. The Modeling Process

The model and operational requirements drove each step in the development of the LH2 FFM. The LH2 FFM is intended for use in a real-time system that automatically detects and isolates faults. Therefore, the model development process was an iterative exercise in which a prototype LH2 model was developed quickly and with relative accuracy so it could be integrated with the real-time software. The model development process that was used for the LH2 system is presented in Figure 1, and the next sections provide more detail about each step in the process.

A. Meet with System Experts

The first step in the model development process is to identify the system designers and operators who will serve as points of contact for identifying relevant documents, answering questions, and reviewing the model. The first meeting with the system experts will be an exchange to explain the purpose of the model and to establish what type of information is required by the modelers. The project leadership and modelers will present high-level information about the project and its goals, as well as an overview of functional fault modeling. The system experts are expected to provide the location of system documentation, whether the information resides in databases, a digital documentation repository, or in hard-copy format. The system experts should begin the process of making the documentation available and establishing the sensitivity of the data. The meeting should also incorporate high-level discussions regarding the system's function and operating modes. Understanding the purpose of the system, the services or commodities it is expected to provide, and how the flow of services or commodities may change during different phases of operation will help the modeler begin to formulate the paramount operational modes and hierarchy of the system. Finally, the modelers will need to begin extracting common or difficult faults to isolate in the system from the experts. These discussions will feed the development of use cases that will aid in model testing and validation later in the process. The exchange of information with system designers and operators may occur in a single gathering, but it will likely consist of a series of meetings. It is preferable that at least one of these meetings be held at the system site. During the walkdown, the system experts are able to talk about failures and their physical effects in the operating environment, and modelers are given the opportunity to visually inspect and document the system with photographs. Seeing the hardware often reveals relationships between components in the system that are not obvious from schematics and documentation, and it familiarizes the modelers with the look and feel of the components they will be modeling.

B. Gather System Documentation

After the initial rendezvous with the system experts, the modelers should begin collecting, organizing and inspecting the documentation associated with the system. The initial review of the documentation will acquaint the modelers with what information is available and where. At a minimum, the models require system schematics and instrumentation, but any of the following documents may be useful during the model development process:

- Mechanical and electrical schematics
- Integrated schematics

- System block diagrams
- Failure Mode Effects Analysis (FMEA)
- Critical Items List (CIL)
- Reliability data
- Operational procedures
- Operating criteria
- Maintenance procedures
- Software specifications
- Instrumentation lists
- Interface documents
- Fault Tree Analysis
- Manufacturer data sheets for components
- Documentation of historical problems and corrective actions

C. Identify System Interfaces and Hierarchy

Once the documentation has been gathered, the first step to begin modeling is to determine where the model fits in the context of the fault isolation system. If the model is standalone, then it will be at the top level of the hierarchy and will not need to accommodate external interfaces with other systems. If the model will be integrated with others, then its place in the system hierarchy needs to be understood. For example, according to the requirements, the LH2 model will be part of a larger integrated model. The LH2 subsystem is one of many in the Ground System, and it is expected to interface with the power, pneumatics, and command and control system in the Ground System, as well as the main propulsion system in the Vehicle System[3]. As a result of these dependencies, the modeler will need to either institute naming conventions for the system operating modes and failures that propagate between subsystems or use the previously established naming conventions[4]. Typically, an Interface Control Document (ICD) would be used to manage the system modes and failures to prevent duplication. After determining its place in the system, the modeler will need to determine the requirements for the level of resolution for fault isolation. In some cases, the end user may need to understand which mechanisms are responsible for a particular failure in which cases the failure modes of the components would be required. In other instances, the end user may only be concerned with isolation to a line replaceable unit (LRU) or some other higher level assembly, and the model will not require resolution to the failure mode level. For the LH2 subsystem, the FDIR requirements dictate several levels of fidelity – failure mode, component and line replaceable unit – depending on the model usage.

D. Identify System Operating Modes

The vast majority of systems, from alarm clocks to motorized vehicles to space shuttles, have several system configurations in which they can operate. In an integrated system, there may be multiple phases of operation defined by the unique configuration of each of its elements. The system configurations and phases of operation are important elements to a FFM because they determine what conditions constitute a fault and how the fault will propagate through the system. The modeler must document these system modes for inclusion in the model. The FDIR project has requirements to model system modes and to be able to switch between the system modes during real-time operation. The LH2 FFM development required system level modes, like vehicle processing, launch countdown and liftoff, to be enumerated first. For each of these phases, the LH2 subsystem-level modes were defined. During the vehicle processing phase, the LH2 subsystem is configured for maintenance and re-fueling. During the launch countdown, the LH2 subsystem undergoes several configuration changes, including the major system modes of chilldown, slow-fill, fast-fill, and replenish. When the components with states, such as valves are relays, in the LH2 system were modeled, their states were defined based on the system modes identified here.

E. Identify and Model System Components and Connectivity

Once high level details of the system hierarchy and system modes are established, the modeling of the system can begin. If none is available, a modeling conventions document should be created before modeling. The modeling conventions document should specify naming conventions for each level of the system hierarchy, color conventions, system mode naming conventions, and failure and test naming conventions. The modeling conventions document may also provide modeling best practices that provide examples of the preferred modeling methods. The conventions document should be developed with the system requirements in mind. In the case of the FDIR project, the requirements for ease of validation by the system experts and scalability for a large integrated model drove the conventions document's best practices.

A brief review of the system schematics and FMEA should reveal common components that will be used in the model, such as transducers, valves, tanks, relays, regulators and power supplies. If the components are used in several places in the system, it is efficient to model each common component according to the model conventions and test its failure propagation and fidelity in a parts library. Once the common components have been modeled, the modeler can select parts from the library to add to the model with the confidence that its failure modes and failure effect propagation paths are correct.

The modeler will then select and name parts from the library, create parts unique to the system, and connect the failure effect propagation paths between failure modes and components. If reliability data is available, it will be added to the components at this stage in the modeling process. The modeler should also review historical failures and corrective action data during this step to ensure that the full scope of failure modes is incorporated in the model.

F. Identify and Model System Instrumentation

After creating the model structure, adding components, and establishing failure effect propagation paths through component connectivity, the next step is to add the system instrumentation that detects failures at different points in the system. Since the FDIR project requires automated fault isolation, the LH2 model included all available analog and digital measurements. However, some project requirements may specify accommodations for guided troubleshooting. In that case, the modeler may need to include observation points besides analog and digital measurements, such as gauges, indicator lights, and test points. In the LH2 model, the analog and digital measurements were obtained from instrumentation lists and electrical schematics, and the tests to detect specific failures from those measurements were defined based on the modeling conventions.

G. Test Model

Once a model with failures and observation or test points is available, the model should be tested to verify its fidelity according to the modeler's understanding of the system. FFM development software provides analysis tools to evaluate aspects of the model, such as testability analysis, forward and backward fault propagation chaining, fault trees, and even real-time analysis to test dynamic failure conditions. Depending on their availability in the development environment, each of these tools should be used to ensure the model is operating as expected. The analysis tools also provide valuable metrics, such as number of failure modes, tests, and fault coverage.

In addition to using the FFM development software tools, it is advantageous to test the model in a relevant development or operational environment. The FDIR project made a priority of having simulated LH2 data and interface software ready to test the LH2 model as early as possible in the development cycle. Having a real-time development environment and an early stage LH2 model was key to the success of the FDIR prototype fault detection and isolation system. The integrated testing helped identify shortcomings of the software and the model at an early stage so that issues in the prototype could be worked in a timely fashion. The real-time test environment also presented an opportunity for the modelers and software developers to coordinate and identify which parts of the software development should be data driven to accommodate model configuration changes and future models. In addition, the real-time test environment provided an opportunity to evaluate the model and software's scalability and performance against the FDIR requirements. Without the test environment, there would be no means of providing realistic performance estimates for the operation of the model in real-time.

H. Review with System Experts

Finally, the modelers should close the loop with the system experts to validate the model that was developed. The model review may include scrutiny of the FFM using the model development software and a demonstration of the model isolating faults in real-time. The FFM review should include formal validation of the system hierarchy, interfaces, system modes, components, failure modes, and connectivity. The system experts should be able to identify system modes that have been incorrectly defined or omitted, components and failure modes whose relationships are inaccurate, and absent failure modes. A real-time demonstration of the fault isolation capability should include use cases as identified during the initial model process step. The real-time demonstration will provide the system experts with a forum for submitting ideas on what type of information should be displayed in a fault isolation application and how the information should be presented. When the model is ready for a final validation review for certification, each step in the review process should be formally documented to provide the required traceability.

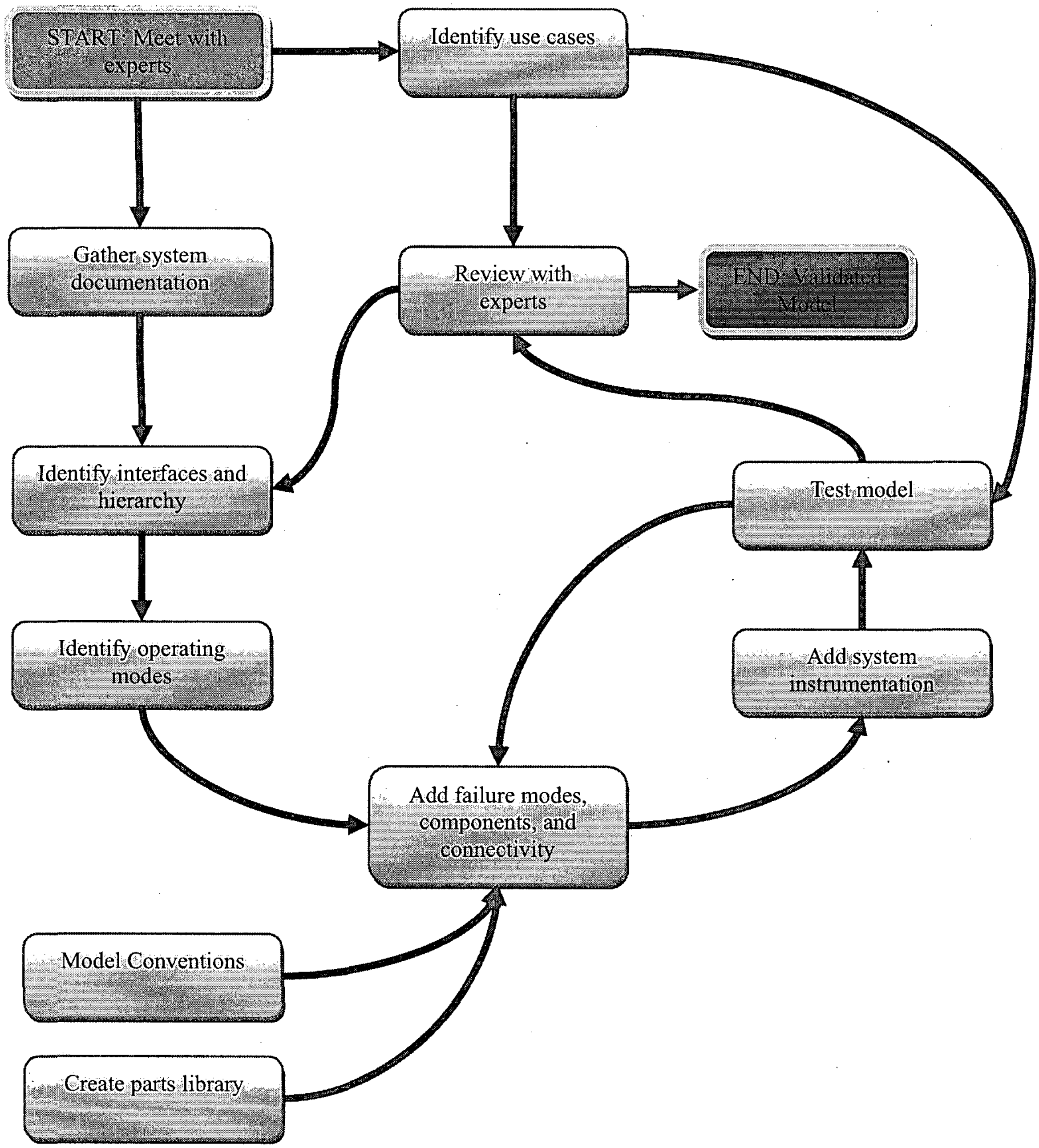


Figure 1. Functional Fault Model Development Process.

V. Results

The result of the focused modeling process was a successful demonstration of the end-to-end system, complete with verified model, software, telemetry interface and graphical user interface. A block diagram of the components of the demonstration is presented in Figure 2. The data file, Python message script, WrapperD application, and Java GUI were all developed specifically for the demonstration. Although the demonstration was very well received by the Ground Operations customer and various end user groups, it does not accurately represent the way fault isolation would be performed as an integrated part of the Launch Control System (LCS). In the future, the FDIR Fault Isolation software will run on an AIX platform and subscribe to events from the LCS message bus that indicate whether a telemeter value has gone outside its range. More testing of the model will be required to understand how fault isolation will be affected by the timing of incoming events. The WrapperD application will be responsible for mapping the events to a test to fail. The WrapperD application will then request the diagnosis from the fault isolation reasoner and publish the diagnosis results back to the LCS message bus, and an LCS display server will be responsible for presenting the diagnosis to the console operator.

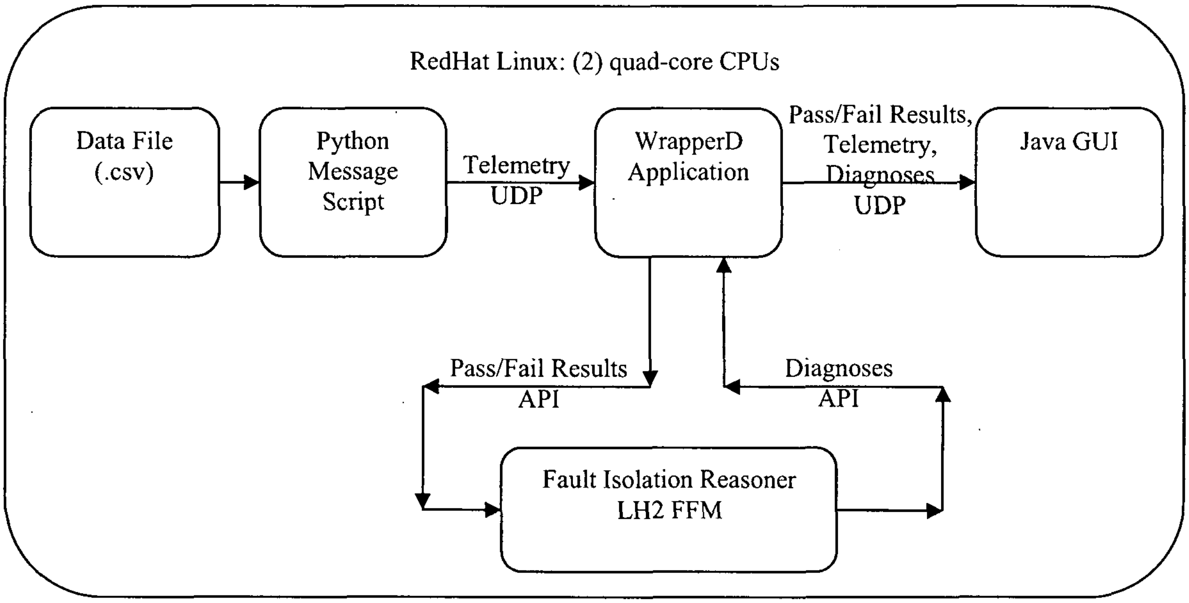


Figure 1. Functional Fault Model Demonstration Components.

VI. Conclusion

The significance of the fault detection and isolation LH2 system prototype is that it provides a framework for future modeling efforts and a real-time diagnostic system. The LH2 system model will continue to be improved with the addition of more components and failure modes and more testing with simulated and/or live data.

Acknowledgments

We would like to thank NASA's Exploration Technology Development Program for their past and future funding of the Fault Detection, Isolation and Recovery project to develop and mature fault isolation technologies for future space missions.

References

[1] Ferrell, B., Lewis, M., Perotti, J., Oostdyk, R., Spirkovska, L., Hall, D. and Brown, B., "Usage of Fault Detection Isolation & Recovery in Constellation Launch Operations," Proceedings of the AIAA SpaceOps 2010 Conference, AIAA, Huntsville, AL, 2010.

[2] S. Deb, S. K. Pattipati, V. Raghavan, M. Shakeri, and R. Shrestha, "Multi-signal flow graphs: a novel approach for system testability analysis and fault diagnosis," IEEE Aerospace and Electronic Systems Magazine, Volume 10, Issue 5, May 1995.

[3] Ferrell, B., and Oostdyk, R., "Modeling and Performance Considerations for Automated Fault Isolation in Complex Systems," Proceedings of the IEEE Aerospace 2010 Conference, IEEE, Big Sky, MT 2010.

[4] Ferrell, B., Lewis, M., Perotti, J., and Oostdyk, R., "Functional Fault Modeling Conventions and Practices for Real-Time Fault Isolation," Proceedings of the AIAA SpaceOps 2010 Conference, AIAA, Huntsville, AL, 2010.

Overview

- Background
- Requirements
- Modeling Process
- Results

Background

- Fault Detection, Isolation and Recovery (FDIR) project is funded by NASA's Exploration Technology Development Program (ETDP)
  - mature fault detection, fault isolation, anomaly detection, and prognostics technologies
  - Constellation Program and future extra-planetary missions
  - Designed to be integrated with Ground Operations
  - automate fault detection and isolation during maintenance and checkout and launch countdown
  - integration of several ISHM capabilities
- The LH2 functional fault model (FFM) was the first model developed by FDIR

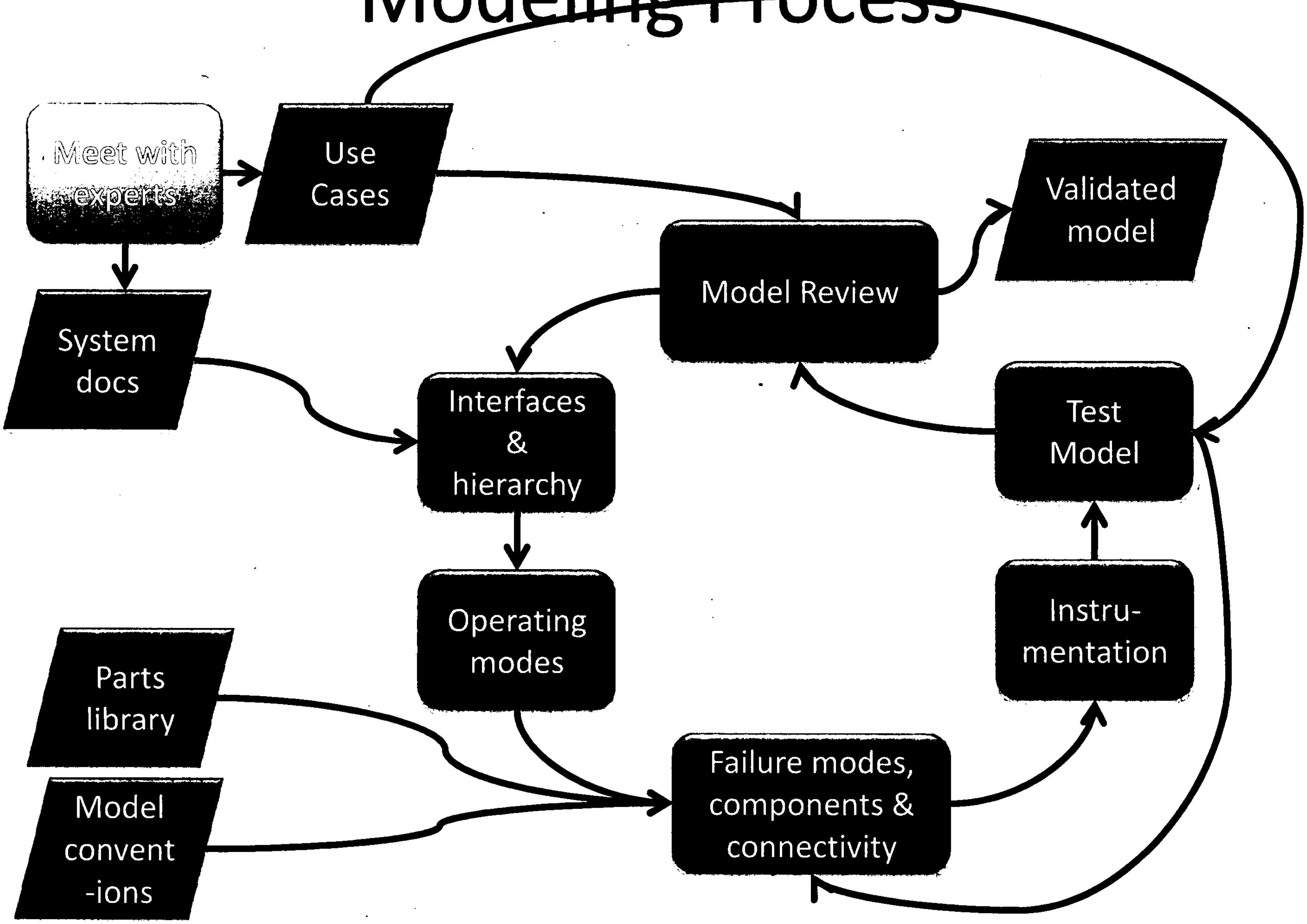
Model Requirements

- Model shall have clear mapping back to physical system to aid in initial model validation by system experts and maintainability and sustainability by system design engineers.
- The model shall be capable of isolating to multiple levels of resolution for the vehicle and ground systems (i.e. failure mode, component, line replaceable unit, etc.).
- Modeling techniques and practices shall be scalable for a large, integrated model that encompasses vehicle systems, ground systems, and facility infrastructure. The integrated model will have an estimated 40,000 failure modes and 50,000 test points if it includes the ground systems, launch vehicle and Orion capsule.

Operational Requirements

- The reasoner shall diagnose multiple independent faults that occur simultaneously.
- The reasoner shall provide a minimal component set based on a particular fault.
- The reasoner shall re-configure the failure effect propagation paths to reflect the current mode of operation and system configuration within one second of a mode change.
- Detected systems faults shall be isolated to the level required for recovery of function.
- Detected faults shall be isolated to the level required for removal of line replaceable unit (LRU).
- Integrated system faults shall be diagnosed to the level required for removal of line replaceable unit (LRU).
- Fault isolation results shall be provided within 1 second of fault detection. The results shall include lists of suspect or bad items from the model.

Modeling Process



Meet with experts

- Identify the system designers and operators
- Explain the purpose of the mode
- What type of information is required by the modelers?
- Present high-level information about the project and its goals
- Overview of functional fault modeling
- Provide the location of system documentation
    - Availability & sensitivity
- System's function and operating modes
- Common or difficult faults to isolate in the system
- Development of use cases
- System walkdown
    - Failures and their physical effects in the operating environment
    - Visually inspect and document the system with photographs

System docs

- Mechanical and electrical schematics
- Integrated schematics
- System block diagrams
- Failure Mode Effects Analysis (FMEA)
- Critical Items List (CIL)
- Reliability data
- Operational procedures
- Operating criteria
- Maintenance procedures
- Software specifications
- Instrumentation lists
- Interface documents
- Fault Tree Analysis
- Manufacturer data sheets for components
- Documentation of historical problems and corrective actions

Use Cases

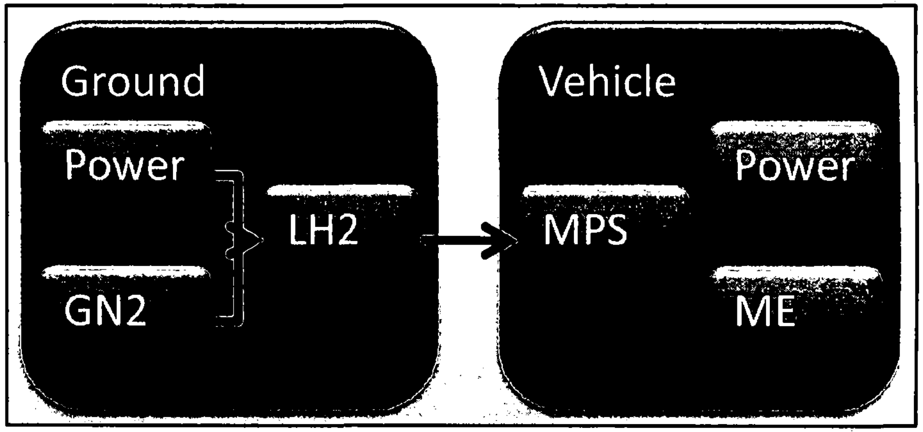
Categories of Failures

- Valve electrical failures
- Valve mechanical failures
- Transducer failures
- Configuration failures
- Maintenance failures
- Failures propagating from other subsystems

Interfaces & hierarchy

- •Model context
  - •Standalone
  - •Integrated
    - •naming conventions for the system operating modes and failures that propagate between subsystems
    - •Interface Control Document (ICD) to manage the system modes and failuresL
- •Level of resolution for fault isolation
  - •failure modes
  - •Components
  - •line replaceable unit (LRU) or assembly

LH2 System



Operating modes

- System configurations
- Phases of operation defined by the unique configuration of each of its elements
- Requirements to model system modes and to be able to switch between the system modes during real-time operation
- System level modes
  - vehicle processing
  - launch countdown
  - Liftoff
- LH2 subsystem-level modes
  - vehicle processing: maintenance and re-fueling
  - Launch countdown: chilldown, slow-fill, fast-fill, and replenish

Model convent-ions

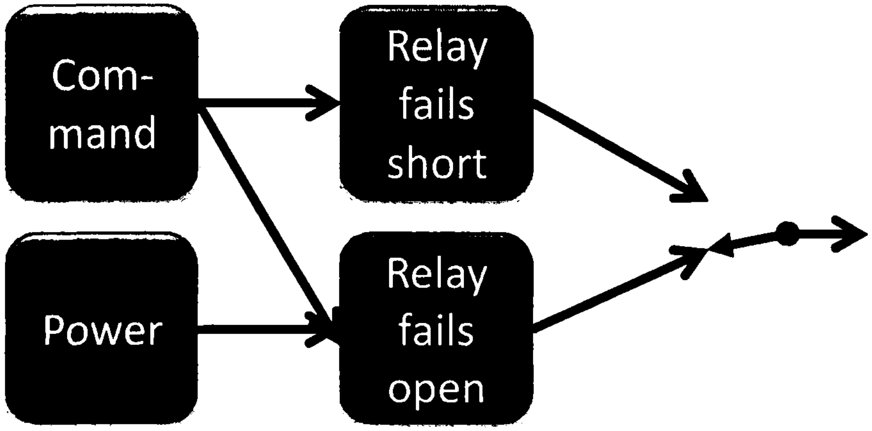
- Modeling conventions
  - Module naming, colors, system mode naming, and failure and test naming
  - Best practices

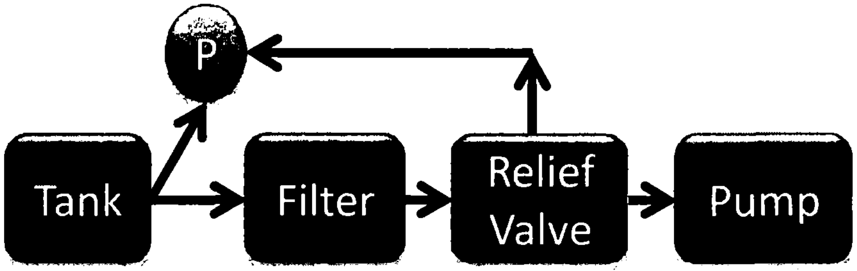
Parts library

- Common components
  - Transducers
  - Valves
  - Tanks
  - Relays
  - Regulators
- Select library parts
- Create unique parts

Failure modes, components & connectivity

- connect the failure effect propagation paths between failure modes and components
- reliability data
- Review historical failures and corrective action data





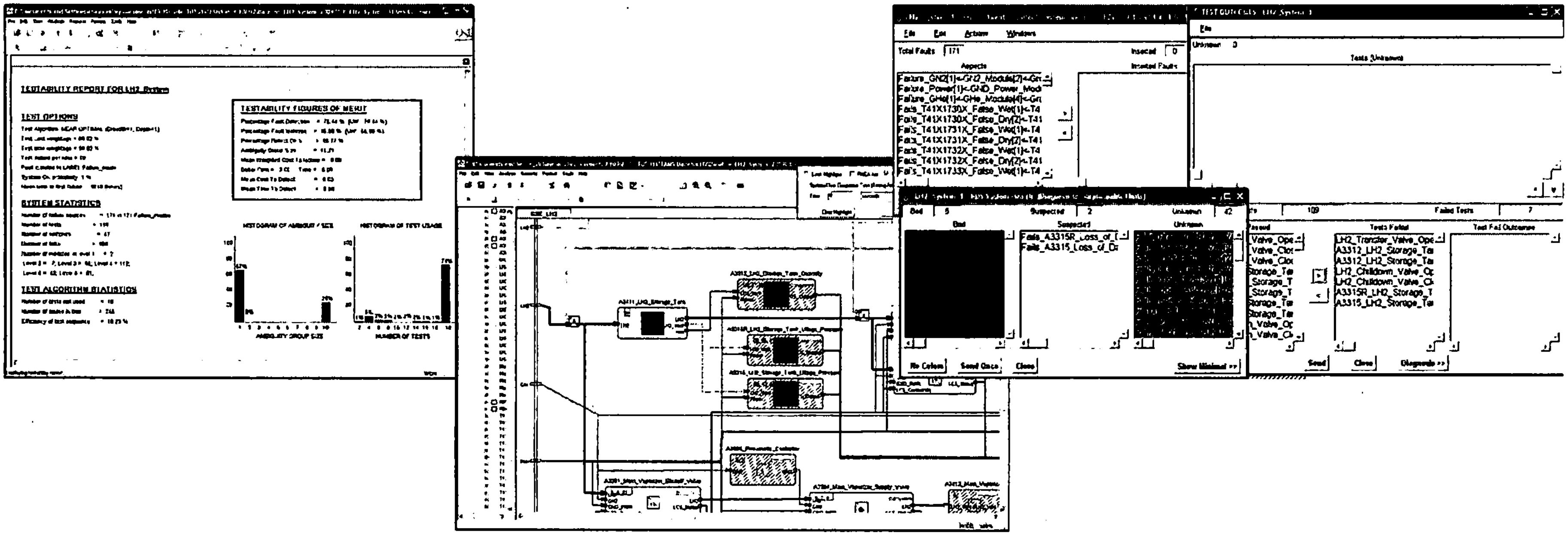
Instru-mentation

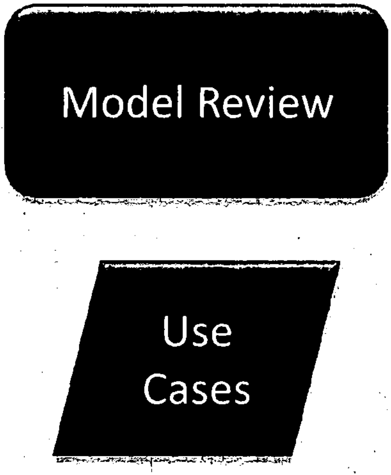
- all available analog and digital measurements
    - Instrumentation lists
    - electrical schematics
- guided troubleshooting
    - Gauges
    - indicator lights
    - test points

Test Model

Use Cases

- FFM development software analysis tools
  - testability analysis
  - forward and backward fault propagation
  - fault trees
  - real-time analysis
- relevant development or operational environment
  - integrated testing helped identify shortcomings of the software and the model at an early stage
  - modelers and software developers to coordinate
  - Identify data driven software aspects for future model changes
  - evaluate the model and software's scalability and performance





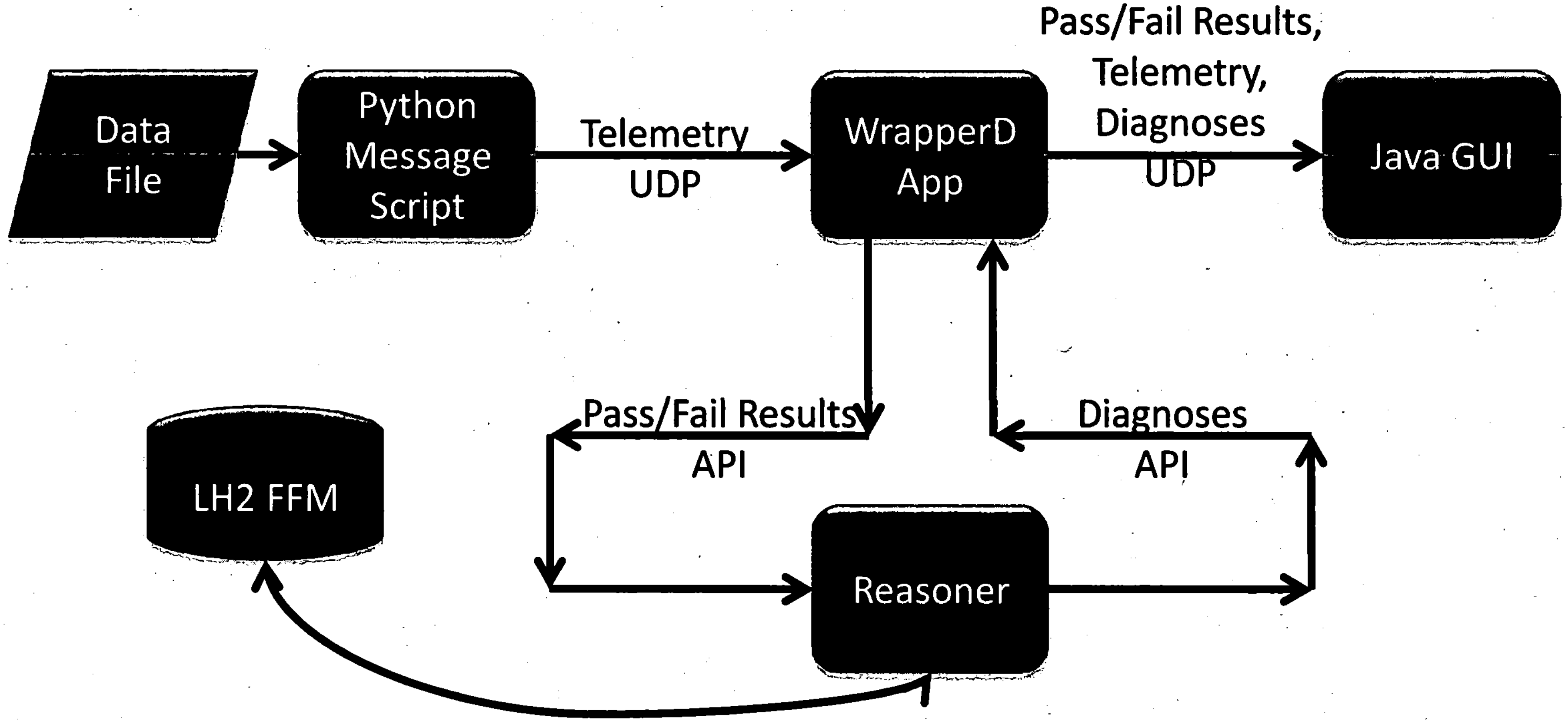
Model Review

Use Cases

- FFM model development software
- demonstration of the model isolating faults in real-time
- formal validation of:
    - system hierarchy
    - Interfaces
    - system modes
    - Components
    - failure modes
    - Connectivity
- use cases as identified during the initial model process step
- Real-time demonstration will provides forum for:
    - what type of information should be displayed
    - how the information should be presented
- final validation review for certification
    - Document each step for traceability.

Results

- Successfully demonstrated the LH2 FFM in a real-time environment



Summary

- Although the demonstration was very well received by the Ground Operations customer and various end user groups, it does not accurately represent the way fault isolation would be performed as an integrated part of the Launch Control System (LCS). In the future, the FDIR Fault Isolation software will run on an AIX platform and subscribe to events from the LCS message bus that indicate whether a telemeter value has gone outside its range. More testing of the model will be required to understand how fault isolation will be affected by the timing of incoming events. The WrapperD application will be responsible for mapping the events to a test to fail. The WrapperD application will then request the diagnosis from the fault isolation reasoner and publish the diagnosis results back to the LCS message bus, and an LCS display server will be responsible for presenting the diagnosis to the console operator.
- The significance of the fault detection and isolation LH2 system prototype is that it provides a framework for future modeling efforts and a real-time diagnostic system. The LH2 system model will continue to be improved with the addition of more components and failure modes and more testing with simulated and/or live data.