Functional Fault Modeling Conventions and Practices for Real-Time Fault 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 @ KSC, Kennedy Space Center, FL, 32899

The purpose of this paper is to present the conventions, best practices, and processes that were established based on the prototype development of a Functional Fault Model (FFM) for a Cryogenic System that would be used for real-time Fault Isolation in a Fault Detection, Isolation, and Recovery (FDIR) system. The FDIR system is envisioned to perform health management functions for both a launch vehicle and the ground systems that support the vehicle during checkout and launch countdown by using a suite of complimentary software tools that alert operators to anomalies and failures in real-time. The FFMs were created offline but would eventually be used by a real-time reasoner to isolate faults in a Cryogenic System. Through their development and review, a set of modeling conventions and best practices were established. The prototype FFM development also provided a pathfinder for future FFM development processes. This paper documents the rationale and considerations for robust FFMs that can easily be transitioned to a real-time operating environment.

I. Introduction

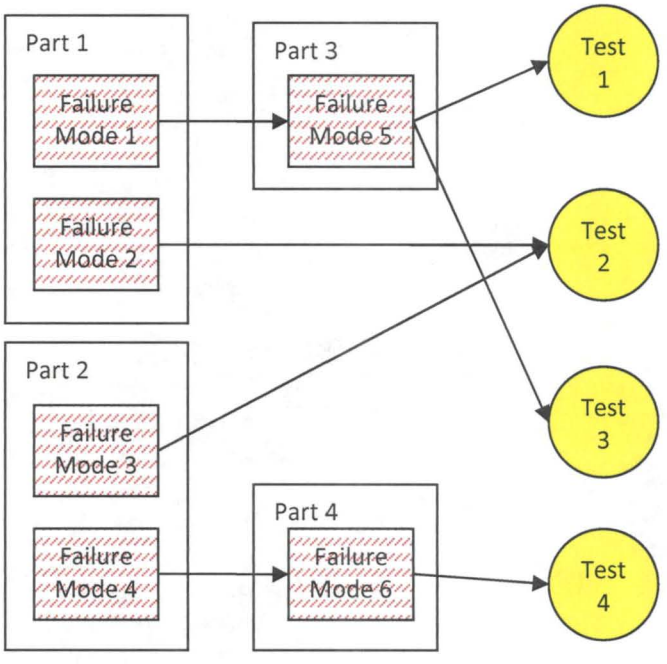
THE Fault Detection, Isolation and Recovery (FDIR) project funded by NASA's Exploration Technology Development Program (ETDP) is purposed to mature fault detection, fault isolation, anomaly detection, and prognostics technologies for use in the new Constellation Program and future extra-planetary missions[1]. FDIR is intended and designed to be integrated with Ground Operations to automate fault detection and isolation during maintenance and checkout as well as launch countdown activities of ground and launch vehicle systems. The FDIR architecture supports the integration of several Integrated System Health Management (ISHM) capabilities, but this paper will focus on the models developed to perform the fault isolation aspect.

Fault isolation is the capability to detect failure conditions in a system and isolate the failure[2] to its root cause. Fault detection methods are wide-ranging and include strategies from simple limit checking of measurements to data mining and statistical analysis to intelligent devices and built-in tests that identify failures at the source. Once a fault or off-nominal behavior is detected, fault isolation techniques are employed to locate the failure mode or modes of the system and implicate bad or suspected components for further testing and replacement. The FDIR project has chosen to employ Functional Fault Models (FFMs) as a means of performing fault isolation. An FFM establishes a relationship between the failure modes of various components in a system with observables or tests that detect those failure modes.

---

[1] Lead 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] ISHM-FDIR Task Lead, NASA-Ames Resident Office, Mailstop ARC

In real-time, a reasoner will evaluate the Dependency Matrix[3], or D-Matrix, that enumerates those relationships to determine which failure modes are likely to be bad or suspect (see Figure 1 for a D-Matrix example). The FDIR project has selected Qualtech Systems, Incorporated's (QSI's) product TEAMS (or Testability Engineering And Maintenance System) as the tool for creating FFMs with a graphical user interface and evaluating the D-Matrix derived from the FFM in real-time. The selection was based on product maturity and certifiability at the time of selection as well as interoperability with other FFMs that were being developed that interface with the Cryogenic System prototype.



Dependency Matrix (D-Matrix)

Failure Mode/ Test	Test 1	Test 2	Test 3	Test 4
FM1	1	0	1	0
FM2	0	1	0	0
FM3	0	1	0	0
FM4	0	0	0	1
FM5	1	0	1	0
FM6	0	0	0	1

Figure 1. Functional Fault Model and Equivalent Dependency Matrix.

The training on the TEAMS tools from QSI and review of other FFMs that had been matured gave ample evidence of the need for modeling conventions. The FFM structure consists of five main components:

1) Modules: representations of failure modes or their hierarchical containers.
2) Arcs: the links connecting modules to one another.
3) Test points: a special type of module that acts as a sink in the model. A test point is the place where the effects of a failure mode can be detected or observed.
4) Switches: devices used to interrupt or redirect an arc.
5) AND nodes: devices that perform a logical AND operation on the arcs. AND nodes are useful for modeling redundancy in a system.

These five components, illustrated in Figure 2, are used to represent the physical structure of the system, and the TEAMS features *functions* and *tests* are used to model the propagation of failures throughout the system. Functions originate at a failure mode, and may be used to represent the effect of a failure in the immediate vicinity or an end effect on the system. Tests reside at the test points, and they represent the calculations that will be performed to determine whether a failure's effect has reached a particular point in the system. The path a function traverses from its originating module, over arcs, through other modules and switches, and ultimately to a test point is called a Failure Effect Propagation Path, or FEPP.

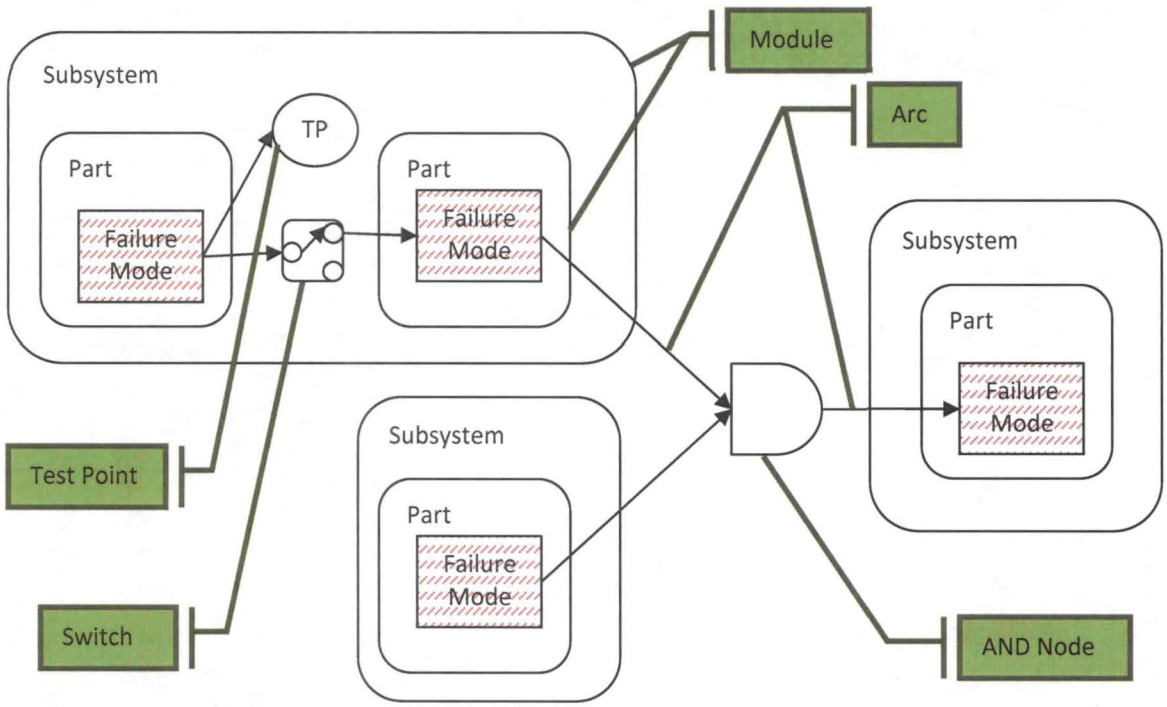


Figure 2. Functional Fault Model Structural Components.

The generic quality of the FFM structural components, functions and tests make the FFM-building environment flexible but introduces several interpretations of how modeling should be carried out. The FDIR project intended for several modelers to complete the task of creating the FFM of the Cryogenic System, and other projects creating the FFMs that would interface with the Cryogenic System employed a different set of modelers. Therefore, it was imperative to create a set of modeling conventions by which all modelers would abide in order to ensure a seamless integration of the models.

Developing model conventions consisted of a process of documenting which features of FFMs were being used by the different modeling groups, requesting QSI's advice on guidelines and best practices, internally reviewing the model conventions and best practices, and organizing meetings that involved all modeling groups to compare and contrast internal modeling practices and agree upon a global set of conventions. In instances where conventions do not allow sufficient flexibility to model all types of failures and their FEPPs, a less restrictive modeling best practice would be suggested. Modeling conventions are generally required where model integration may be affected and modeling practices are relevant for modeling tasks that may be accomplished in several ways without affecting how the FFM will be integrated with another.

II. Modeling Conventions

The modeling conventions agreed upon for FFM of a Cryogenic System are presented in this section and are divided into categories based on whether the convention is generic or specific to the TEAMS tool.

A. General Functional Fault Modeling Conventions

The general FFM conventions should be applied to all FFM developments, regardless of whether the model was created using the TEAMS graphical user interface or some other tool.

1) All FFMs shall use common delimiters between fields and texts in a field to facilitate batch editing and software tools.
2) FFMs that are required to integrate with other FFMs shall agree on unique names for the operational configuration and mission phases of the system.
3) The FFM shall nest modules using the hierarchy labels defined by MIL-HDBK-505 Handbook for Definitions of Item Levels, Item Exchangeability, Models and Related Terms[4]. The following hierarchy labels and field formats shall be used by the Cryogenic System FFM.

a. Failure mode: The manner by which a failure is observed and describes the way it occurs. A failure mode shall be formatted as follows: <Text-description>_[<FMEA-ID>], where the FMEA-ID field is optional.
b. Part: One piece or two or more pieces joined together which are not normally subject to disassembly without destruction of designed use[4]. Examples of parts include orifices, limit switches, resistive thermal devices, thermocouples, resistors, capacitors, strain gauges, and accelerometers. A module with a "part" hierarchy label shall be formatted as follows: <Text-description>_<Part-number>.
c. Assembly: A number of parts joined together to perform a specific function or unit replaceable as a whole that are capable of disassembly into one or more replaceable parts[4]. Examples of assemblies include fan assemblies, pressure relief and check valves, transducer and signal conditioners, quick disconnects, filters, tanks and dewars, circuit cards, and vaporizers. The name of an assembly module shall be formatted as follows: <Text-description>_<part-number>.
d. Unit: An assembly or any combination of parts, and/or assemblies mounted together normally capable of independent operation in a variety of situations[4]. For example, pneumatically actuated cryogenic valves, motors, engines, pumps, power supplies, electro-pneumatic controllers, and auxiliary power units are considered "units." The name of a unit module shall be formatted as follows: <Text-description>_<part-number>.
e. Group: A collection of units, assemblies, or parts connected together or used in association to perform an operational function[4]. Examples of groups include Ground Hydraulic Power Units and Hydraulic Control Units. The name of a group module shall be formatted as follows: <Text-description>_<part-number>.
f. Subsystem: A combination of groups, etc., that performs an operational function within a system and is a major subdivision of the system[4]. Subsystem examples include Liquid Hydrogen (LH2), Liquid Oxygen (LOX), Ground Power, and Pneumatics. Modules with the "subsystem" hierarchy label shall be formatted with a unique system identifier (see next definition), followed by the subsystem acronym or description, according to the format: <system-identifier>_<subsystem-acronym>. For example, the Ground Liquid Hydrogen subsystem would bear the name GS_LH2 which stands for "Ground System Liquid Hydrogen".
g. System: A system is a combination of subsystems that can be considered as a self-sufficient unit in its intended operational environment[4]. For example, a system could be the Ground System or Vehicle System. "System" module labels shall contain a text description of the system in the following format: <Text-description>.

4) Tests in FFMs shall conform to the following four categories to facilitate their use in the Fault Isolation application code development.
a. Consistency: Parts with two states, such as valves and relays, shall have two tests per indicator to verify all possible conditions. The consistency test type performs logical operations that determine whether a dual-state part is good or has some failure condition. The real-time test logic will compare the commanded state of the part to its indicators. FFM consistency test names shall reflect the part name, indicator type (open or closed/on or off), the indicator's measurement identifier, and a text description of the failed consistency check in the following format: <part-name>_<indicator-type>_<measurement-identifier>_<text-description>.
b. Discrete: A discrete part shall have two tests to verify all possible conditions. The discrete test type verifies the state of a discrete part is properly reflected by its indicator during a particular system mode. FFM discrete test names shall reflect the part name, indicator type (hi or lo/on or off/wet or dry/etc.), the indicator's measurement identifier, and a text description of the failed discrete test in the format: <part-name>_<indicator-type>_<measurement-identifier>_<text-description>.
c. Analog: Depending on the nature of the part and data acquisition system, an analog instrumentation part shall have either three or five tests to verify whether the value is out of range. If the instrumentation's operating range is smaller than the actual range of the transducer, there will be five analog tests for the instrumentation: off-scale low, off-scale high, off-nominal low, off-nominal high, and loss of data. An off-scale test evaluates measurements outside the operating range of the instrumentation but within the range of the transducer. An off-nominal test looks for values within the operating range of the instrumentation that exceeds an engineering limit for the system during a particular system mode. A loss of data test looks for a communication problem between the transducer signal and the location where the measurement is used. If an instrumentation part's operating range is the same as the range of the transducer, there shall be three analog tests: off-nominal low, off-nominal

high, and loss of data. FFM analog test names shall include a text description of the measurement and measurement type, the indicator's measurement identifier, and a text description of the test in the format: <measurement-description>_<measurement-type>_<measurement-identifier>_<test-description>.

d. Custom: Custom test types do not fit into the consistency, discrete or analog test type. These tests are more complex and involve sensor fusion or complex mathematical operations such as filtering and frequency analysis. FFM custom test names shall include a text description of the measurement, measurement type, the indicator's measurement identifier, and a text description of the test in the format: <measurement-description>_<measurement-type>_[<measurement-identifier>]_<test-description>.

B. TEAMS-specific Functional Fault Modeling Conventions

The remaining modeling conventions specifically apply to the TEAMS FFM implementation that was selected. The purpose of these modeling conventions is to optimize some aspect of the TEAMS FFM for model validation, sustainability, or real-time performance.

1) Parts with discrete states shall be modeled using switches in TEAMS. The switches shall represent the parts in their normal state. For example, a normally closed valve would be represented with a switch whose failure effect propagation path is normally broken.
2) The system mode feature in TEAMS shall be used to represent different ground and vehicle configurations and phases of vehicle processing.
3) The scope of function propagation along failure effect propagation paths shall be managed with naming conventions.
   a. Generic: Functions that propagate between subsystems shall use the generic naming convention. The application of a generic function name is demonstrated in the case of a loss of ground power. Multiple units, assemblies, and parts in the ground and vehicle systems may be supplied by the ground power. A generic function for the loss of ground power can be detected by tests in many subsystems without the need to know which specific part in the ground power system caused the failure. Every part, assembly, unit, etc. in the ground power subsystem that may result in a loss of ground power should bear the generic function. The generic function name shall start with the prefix "GEN" followed by the subsystem where the failure originated, and a description of the failure, as in <GEN>_<subsystem-name>_<text-description>.
   b. Specific: Functions relevant to a specific subsystem that do not cross subsystem boundaries shall use the specific naming convention. The specific functions will only be detected by tests local to their subsystem. The specific function name shall have a prefix that indicates its subsystem followed by a text description of the failure mode as identified in the FMEA documentation. The format for specific functions shall be: <subsystem-name>_<text-description>.
4) Hardware parts, assemblies or units that provide analog or discrete data feedback and/or measurements derived from physical hardware that provides failure effect insight shall have a test point at or near the modeled item. Test point labels in TEAMS shall begin with the schematic identifier, followed by the sensor type and then "TP" in the following format: <schematic-id>_<sensor-type>_<TP>.

III. Modeling Best Practices

Functional fault modeling of the Cryogenic System also revealed some modeling options that did not necessarily need to be imposed as conventions; rather they were added as modeling best practices to maintain a consistent look and feel between FFMs. The following sections give the recommended modeling best practices for different situations.

A. Bidirectional Flow

In some systems, particularly fluid systems, it is necessary to model bidirectional flow of a commodity. QSI presented the modelers with two viable alternatives for representing bidirectional flow – switches and functions. The switch method for representing bidirectional flow involves adding TEAMS switch components around each module in the FFM that allows flow to occur in both directions. The switches each have two states – one for flow in the left-to-right direction, and one for flow in the right-to-left direction. A system mode will dictate which direction is currently active for all switches. The switch solution is the preferred solution because it most closely follows the other modeling conventions and best practices related to switch and system mode use without interfering with

function naming conventions. The switch solution also has the advantage of being easier to visualize, which makes model validation and sustainability less costly. The biggest disadvantage to the switch solution is the addition of more switches to the model, which results in a larger model memory footprint and real-time performance degradation during system mode changes. An example of a FFM using switches to implement bidirectional flow is shown in Fig 3.

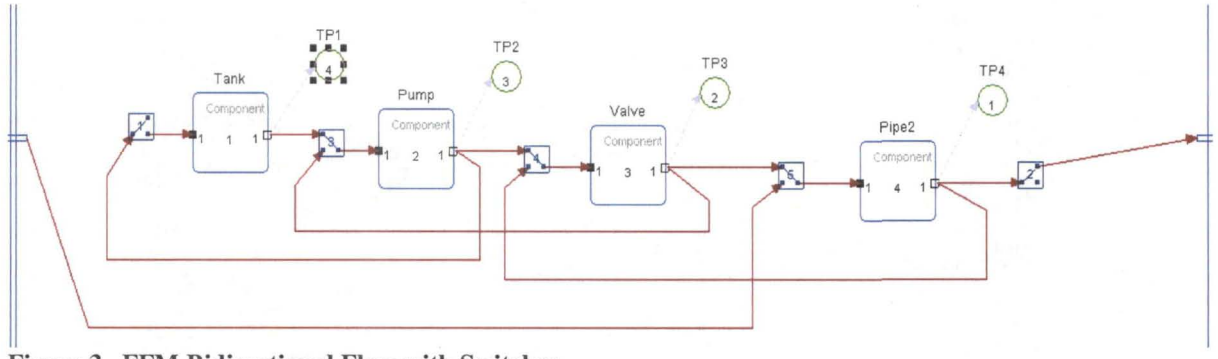


Figure 3. FFM Bidirectional Flow with Switches.

The other bidirectional flow mechanism presented by QSI is the use of functions. In this implementation, a test point is placed in the model for each direction of flow. The test points each have multiple tests representing the progression of the function propagation from the first part to the part just before the test point. For example, at one test point, the first test only detects the function that originates at the first part. The second test only detects the functions originating from the first two parts, etc. In the second test point, the first test only detects the function that originates at the last part. The second test only detects the functions originating at the last part and the second to last part, etc. In real time, only the tests at the test point corresponding to the current flow direction would be evaluated. An example of an FFM with bidirectional flow implemented in functions is presented in Figure 4. Although modeling bidirectional flow with functions is functionally equivalent to the use of switches, it masks the details of the implementation in the real-time software and makes model validation more difficult.

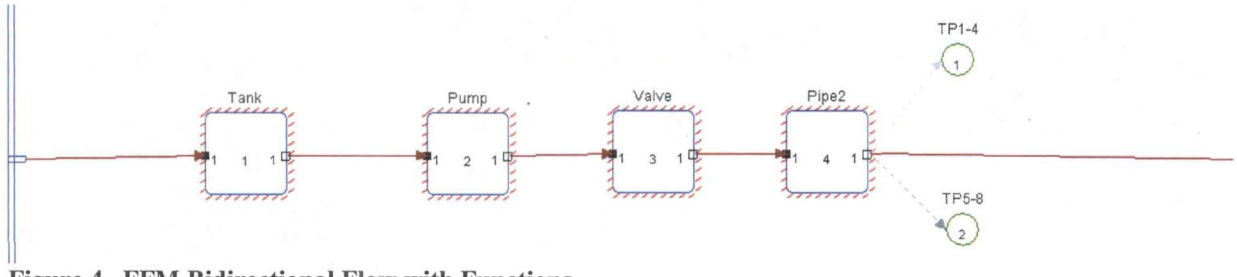


Figure 4. FFM Bidirectional Flow with Functions.

B. Controlling Failure Effect Propagation Paths

Where possible, switches should be used to control Failure Effect Propagation Paths (FEPPs). By convention, switches are used for modeling discrete-state parts and for representing system configurations and mission phases, but switches are also useful for controlling FEPPs where there are feedback loops and other propagation paths that may inadvertently allow functions to propagate to undesirable locations. For example, feedback loops or bidirectional paths may allow a function to propagate through a valve assembly, even though the valve is closed and its associated switch state is meant to prevent any functions from propagating through it. In this event, a switch may be used at the entry to the valve assembly on the feedback loop or bidirectional flow arc to keep stray functions out of the valve assembly when it is closed. The switch will have the same discrete states as the switch that is meant to model the open and closed operation of the valve.

C. Color Conventions

Colors for hierarchy levels and arcs (links) between modules should follow color code standards where relevant. In the Cryogenic System, the line color of arcs follows the KSC Safety Standard for Ground Piping Systems Color

Code and Identification[5]. Following a KSC safety standard for piping systems eliminates confusion and allows the modelers and system experts to efficiently follow the flow propagating to and from modules.

D. Failure Rates and Probabilities

Most functional fault modeling tools make concessions for including failure rate data in the FFM so that suspected failures may be ranked according to the probability of occurrence. However, this failure rate data is not always readily available. In the absence of complete or reliable failure rate data, parts may instead be grouped into probability classes that collect like parts with similar failure probabilities. For the Cryogenic System FFM, five failure probability classes were chosen.

1) High Probability, such as transducers
2) Medium-High Probability, such as rotating components and leaks
3) Medium Probability, such as valves and dynamic components
4) Medium-Low Probability, such as the data acquisition system
5) Low Probability, such as structural components

E. Modeling Passive Parts

Manual valves, gauges and test ports shall not be modeled unless specified by the responsible design and/or system engineer. Modeling manual valves, gauges, and test ports does not aid automated fault isolation unless downstream instrumentation exists that can isolate the fault. However, the design engineers or system experts may request to add certain components to the model that may provide benefits regarding operations, troubleshooting, or fault isolation.

F. Model Parts Library

A TEAMS library of generic parts should be created and used to populate the parts of a subsystem. The generic parts library standardizes the look and feel of common parts in different subsystems and saves modelers' time from developing parts from scratch. A generics parts library also gives new model developers a starting point for understanding how parts can be represented in an FFM and familiarizes them with common parts and their failure modes.

IV. Functional Fault Model Development Process

In addition to acting as a pathfinder for modeling conventions and best practices, the FFM for the Cryogenic System illuminated the required steps in the model development process. This process will be refined and used as a guide for future FFM development. Figure 5 is a graphical depiction of this model development process as was developed during the Cryogenic System FFM development.

V. Conclusion

In summary, a prototype functional fault model of a Cryogenic System was developed using COTS TEAMS software. The FFM building effort exposed several model variations and issues that were resolved by agreements among modeling groups at NASA to abide by model conventions and best practices. In addition to fueling the development of conventions and best practices, the prototype Cryogenic System FFM helped identify the major steps in the FFM development process.

One of the most important lessons learned from the model conventions development process was the need for an overseer who could resolve differences between modeling teams and make the ultimate decision about model conventions when there is an impasse. Although most model conventions were quickly adopted, there were a few that required one group of modelers or another to have to retroactively change their models. The number of models affected often drove the decision for which modeling convention would be adopted, but a better solution would be to resolve the conflicts based on the usability, performance, and sustainability of the models. Another lesson learned during the model conventions development was the need to consult existing standards for modeling. During the course of the reviewing and agreeing upon model conventions, the model groups discovered military and NASA standards that applied to FFM development and aided in resolving modeling conflicts between the groups.

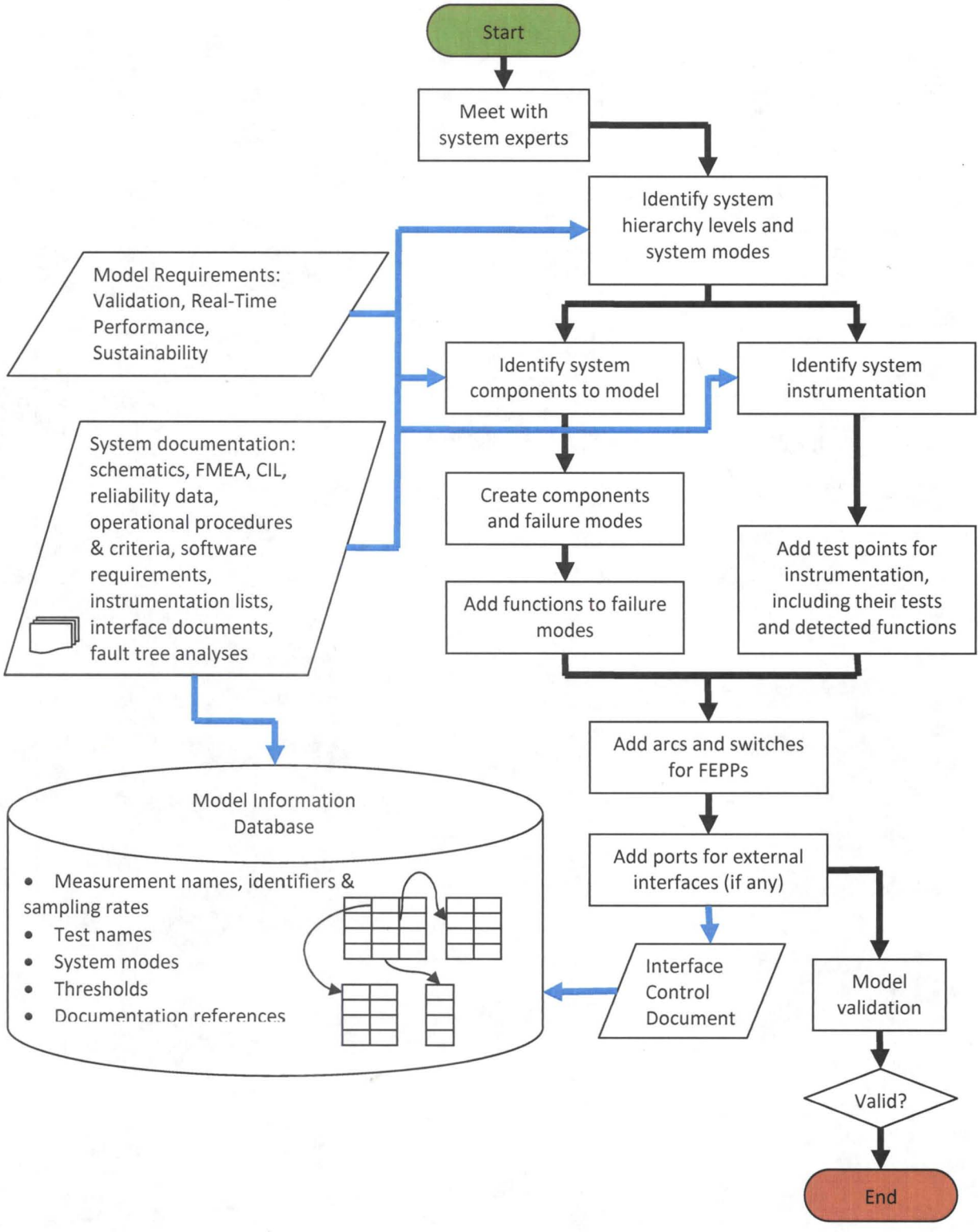


Figure 5. FFM Development Process.

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]Heimerdinger, W. L., and Weinstock, C. B., "A Conceptual Framework for System Fault Tolerance," USAF CMU/SEI-92-TR-033, ESC-TR-92-033, 1992.

[3]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.

[4]"Handbook for Definitions of Item Levels, Item Exchangeability, Models and Related Terms," MIL-HDBK-505, 1998.

[5]"Safety Standard for Ground Piping Systems Color Coding and Identification," KSC-STD-F-0004, Revision B, 1982.

Functional Fault Modeling Conventions and Practices for Real-Time Fault Isolation

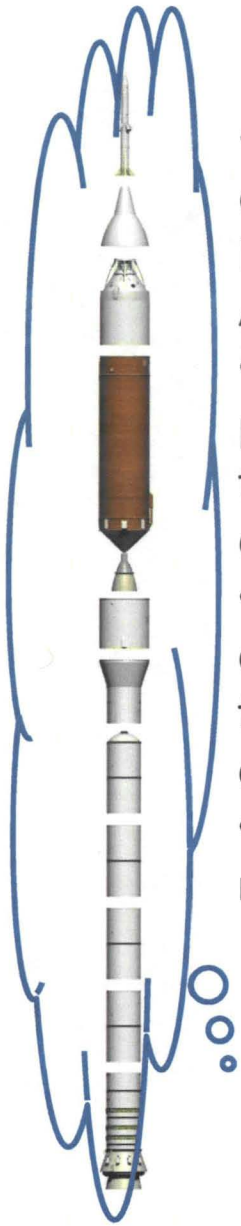
Presented by: Bob Ferrell

AIAA Space Ops Conference
Huntsville, AL
April 26-30, 2010

Overview

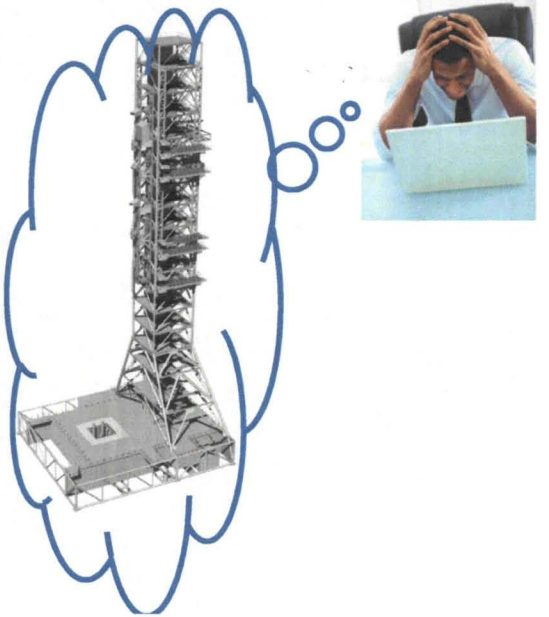
- Problem Background
- Functional Fault Modeling
- The Need for Modeling Conventions
- Modeling Conventions and Best Practices
- Lessons Learned

Background



- Model originally developed for Functional Fault Analysis
- Capture failure propagation times of critical conditions
- Analyze fault coverage for faults of different criticalities
- Not intended for real-time use

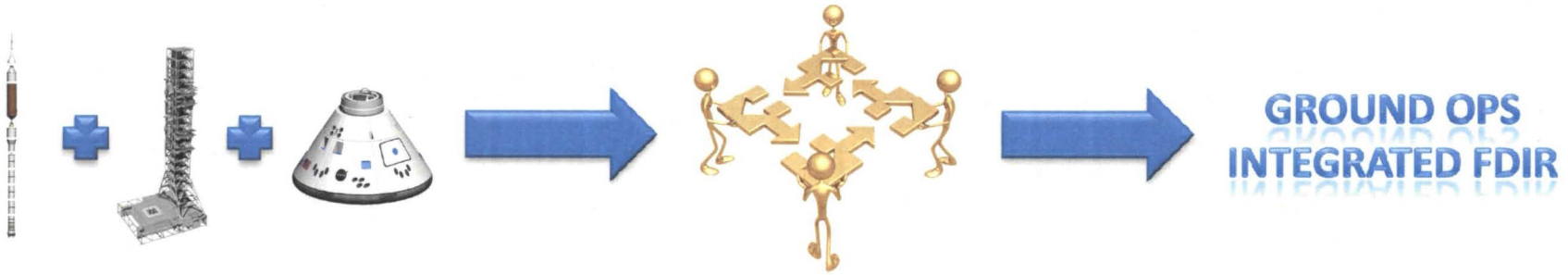
- Model developed to evaluate results of BITs
- Intended for on-board use



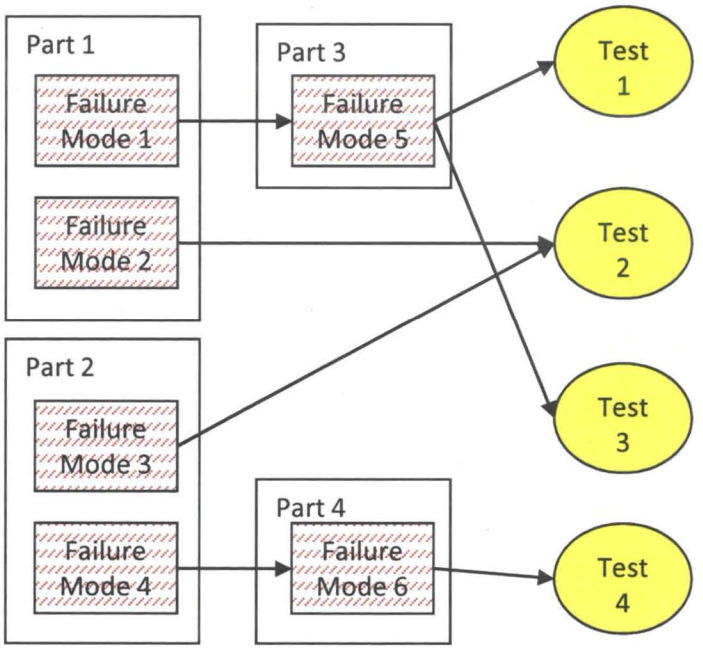
- Proof-of-concept model developed for Ground Operations Launch Control System (LCS)
- LCS opted to include concept in their baseline
- Intended for real-time use

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
  - automate fault detection and isolation during maintenance and checkout and launch countdown
  - integration of several ISHM capabilities
- When FDIR was adopted by Ground Ops LCS, the integration of Ground, Ares and Orion models had to be considered

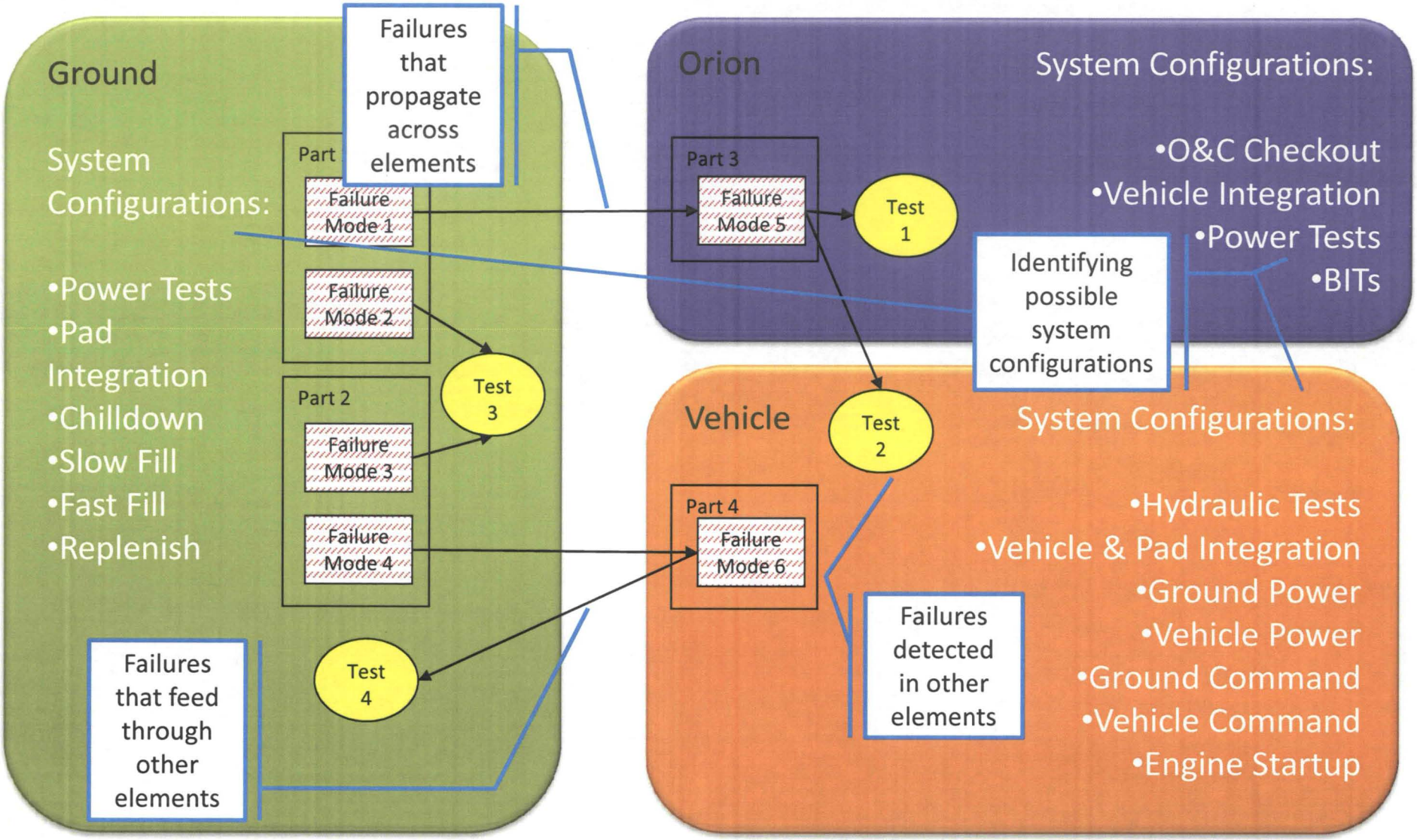


Function Fault Modeling



Failure Mode/ Test	Test 1	Test 2	Test 3	Test 4
FM1	1	0	1	0
FM2	0	1	0	0
FM3	0	1	0	0
FM4	0	0	0	1
FM5	1	0	1	0
FM6	0	0	0	1

FFM Issues for Integrated FDIR



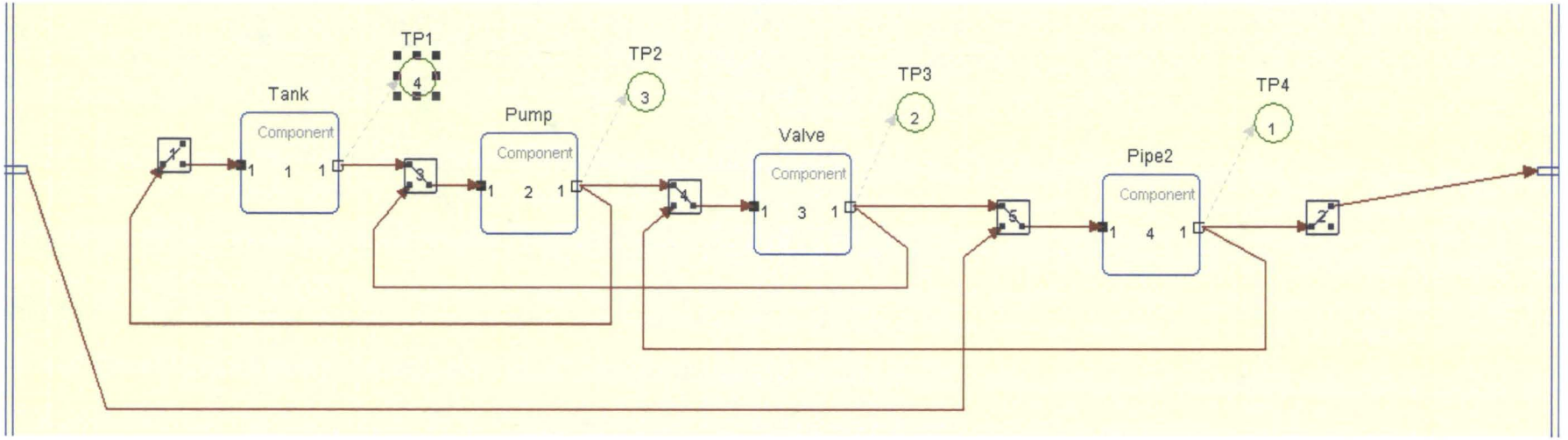
Modeling Conventions

- Module and Test Naming
- System Mode Naming
- Hierarchy Levels and Naming
  - Failure mode
  - Part
  - Assembly
  - Unit:
  - Group
  - Subsystem
  - System
- Tests and Naming
  - Consistency checks - logical operations comparing the commanded and actual position of a dual-state part (such as a valve or relay)
  - Discrete – verifies a discrete sensor is in its proper position during the specified System Mode
  - Analog – tests off-scale low, off-scale high, off-nominal low, off-nominal high, and loss of data conditions
  - Custom – tests that do not fit into the consistency, discrete or analog test categories, such as filtering and frequency analysis

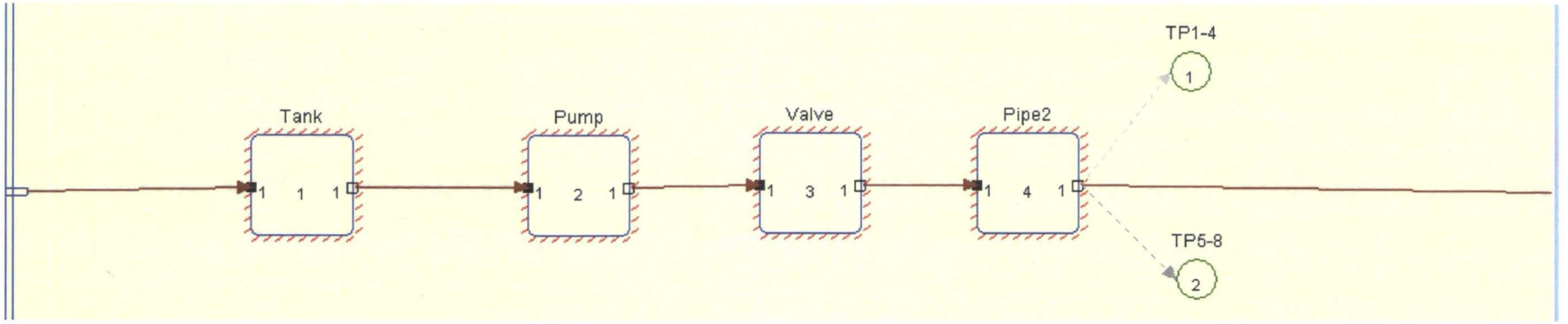
Modeling Conventions

- Discrete states shall be modeled using switches
- Function propagation along failure effect propagation paths is managed with naming conventions
  - Generic or Global - functions that propagate between subsystems
  - Specific or Local - functions relevant to a specific subsystem that do not cross subsystem boundaries
- Test point location

Modeling Best Practices – Bidirectional Flow



Switches vs. Functions



Modeling Best Practices

- Control Failure Effect Propagation Paths with switches
- Apply relevant color code standards
  - Standard for Ground Piping Systems Color Code and Identification
- Where failure rates are not readily available, group failure modes into failure rate categories
  - High Probability: transducers
  - Medium-High Probability: rotating components and leaks
  - Medium Probability: valves and dynamic components
  - Medium-Low Probability: data acquisition system
  - Low Probability: structural components
- Disregard passive parts
- Develop a model parts library

Lessons Learned

- Establish agreements among modeling groups to abide by model conventions and best practices EARLY
- Identify an overseer to resolve differences between modeling teams
  - Models with more maturity may force other models to abide by modeling conventions that aren't ideal
  - The mediator should consider usability, performance, and sustainability of the models
- Consult existing standards for modeling
  - Review military and NASA standards that apply to FFM development
  - Independent source to resolve conflicts between modeling groups
- Result: a prototype functional fault model of a Cryogenic System was developed using COTS TEAMS software
  - The prototype will be integrated with other Ground System and Vehicle System models to yield an Integrated FDIR capability