

MBSMAI Phase 1 Recommended SysML/MagicDraw Modeling Techniques & Tips

In a SysML model, all elements of the system are represented by “blocks,” and it’s the specification of each block that defines its characteristics and role in the model. To create a SysML/MagicDraw model that will support Reliability analysis using Tietronix plugins, the modeler should first **create folders (a.k.a. a “Package”) for every system (Step 1)**, subsystems and components, and define other model elements within that package. This is especially important for larger models and helps with having an organized model structure.

The easiest and best approach to add each element to the model is to right-click on the folder of interest in the containment pane and select the required element from the list of proposed options.

The main element that forms a model in MagicDraw are the “Blocks.” These “block” elements are used to represent systems, subsystems, and components. Block elements are then populated with further information to represent the characteristic of that element. Each block can be a representation of a component itself, its function, or its effect in different failed and operating states.

It may be helpful to create a Block Definition Diagram of the system first to assist the modeler in ensuring that all the system elements are represented by a package.

For example, the Sounding Rocket SysML/MagicDraw model consists of a Block Definition Diagram (BDD), which defines the “Ownership” of the created blocks and shows the hierarchical relationship of blocks. As depicted in Figure B1, Sounding Rocket is the “system block” that owns the Celestial Attitude and Control Subsystem (ACS), Telemetry System, Parachute Recovery, Rocket Engine, and instruments payload as the immediate lower level components in the hierarchy. The Rocket Engine itself consists of 3 major elements: Stage 1 Engine, Stage 2 Engine, and Boost Guidance.

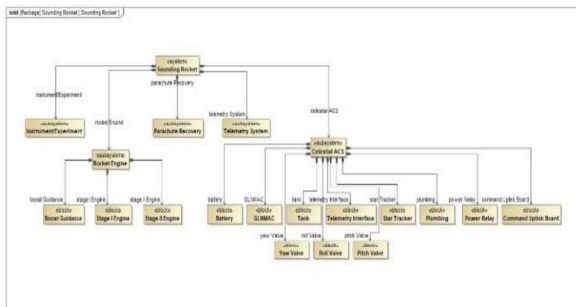


Figure B1: SR Block Definition Diagram

Celestial ACS was the main subsystem that was modeled and reviewed in this case study example and, as seen in Figure B1, it has been modeled one level further into the component level. Eleven components are identified as part the celestial ACS subsystem,

which includes the Airborne Computer (GLNMAC), Command Uplink, Star Tracker, Telemetry Interface, Battery, Power Relay, High Pressure Tank, Valves and Plumbing.

Next the modeler should **configure /“stereotype” each block according to the input needs of the Tietronix plugin (step 2)** or those of the Reliability plugin of choice. Specifically, the JSC provided SysML profile package, “NASA_Profile” which includes common stereotypes of “Function,” “Effect,” and “FailedState,” should be uploaded to the model to enable the Tietronix Plugins. These stereotypes are used in the model to define failures and failure effects that are used by the Tietronix Plugins in the generation of Fault Tree Analysis (FTA) and Failure Mode and Effects Analysis/ Failure Mode Effect and Criticality Analysis (FMECA) Criticality Analysis (FMEA/ FMECA) outputs. While “Signals” are used to represent failure causes.

Although these stereotyped fields can be added to blocks in any order, having a systematic approach will increase efficiency and consistency to the modeling procedure. To add a stereotype, the modeler can either right-click the block, select “stereotype” from the list, or double-click the block and add the stereotyped field.

The recommended order to add these stereotyped field blocks is to define the component “Function,” “Effects,” “FailedStates,” and then “Signal” elements under every state machine to represent failures. To do so, as for other model elements, the modeler should right click on the component of interest, select “Add element” and chose the block from the listed elements; then apply the desired stereotype (e.g., Function) to this block by searching the stereotypes list or by drag-and-drop approach from the “NASA_Profile” package. Functions should be named uniquely, as they are eventually allocated to the unique components owning “FailedState(s)” and used to populate the FMEA Function column.

This Stereotyping procedure should then be used to populate the “Effects” and “FailedState” data as well as all other any other desired modeling data. The “Effect” blocks should also be created, named and stereotyped accordingly. The effects should include the status of the component in both the operating and failed states. These effects will be used later on in the State Machine Diagrams via the “allocation method” to represent the states of the system. The “FailedState” stereotypes are directly applied to the State Machine Diagram which will be explained later. The name of the “FailedState” block should uniquely (e.g., use “Battery Short” not just “Short”) represent the failure

mode, and the allocated effect will be used to fill the failure effects column in the FMEA report.

One last step in this stage is to define “Signals.” Broadcasted “Signals” actions are modeled as entry actions to states and must be unique to each component. Each unique “Signal” is then used as an event trigger on transitions to another component’s state machine. These “Signals,” if appropriately assigned, will later be utilized in the FMEA report (as the failure causes) and FTA analysis (as basic or intermediate events). This process should be repeated for each configuration element. However, copying and/or deleting an element or symbol should be avoided to adhere to uniqueness rules and to avoid software and/or modeling errors, as noted in section 4.2, that will impact the model’s results.

Now, in order to be able to utilize the embedded information and extract relative information from the model, the relationships of and between these operating and failed states should be defined. To do so, the modeler should **create State Machine Diagrams (SMDs) (Step 3)** for every component and for every transition level (i.e., between components and subsystem, or between subsystems and the system) to define the hierarchy of failures throughout the model.

In different model setups, whether stand alone or orthogonal, signals are used as an informative chain to provide the status of that component, which will eventually trigger the next level effect (or subsequent component in a series set up) status.

For example, as shown in Figure B2, a folder (package) was created for the ACS battery, a block was created to represent the battery as a component, and subsequently battery function, in this case; “To provide required power to ACS components” was defined and this block was stereotyped as a “Function,” then corresponding “Effects” of the battery were defined in the same manner in relation to all operating and failed states. For instance, in a normal operating state, the battery is “Operating nominally,” while it can have multiple failed states, such as “Cannot store power in failed string,” “No connection of power lines,” “Reduction of max derated current carrying capacity,” and “Short across connections.” As can be seen in the Figure B2, all these operating states are stereotyped as “Effects.”

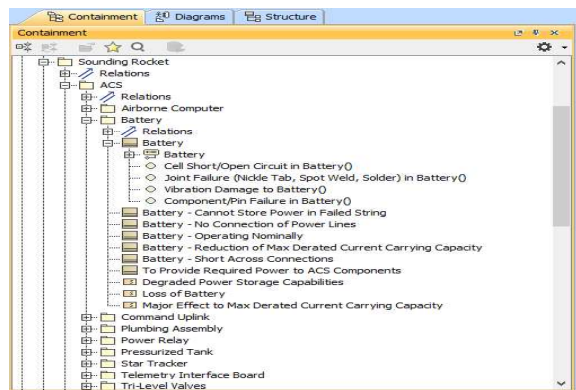


Figure B2: Battery Package

As shown in Figure B2 and Figure B5, signals are elements that basically will be used to provide the state of that component to upstream or upper level parts of the model. As an example, the signal “Loss of Battery” in Figure B2, is assigned to the battery failed state of – power harness failed open – (Figure B5) with an entry named “Battery lost.” The function of this entry is to broadcast this signal when the battery is at this specific state of failure.

State Machines creation is initiated using the same process as adding other elements to the model. When the SMD block is created, the modeler should use the SMD toolbar (Figure B3) to draw the SMD. The modeler should drag and drop a “State” from the toolbar for every state of the system/component. This means every operating, standby and failed state. To distinguish failed states, the modeler should right-click on the state and assign the “FailedState” stereotype to that block. Next, “Transition” lines should be used to connect to these states in order to define the flow of event that would result in transition of the system from one state to another (i.e., from operating state to a failed or vice versa).

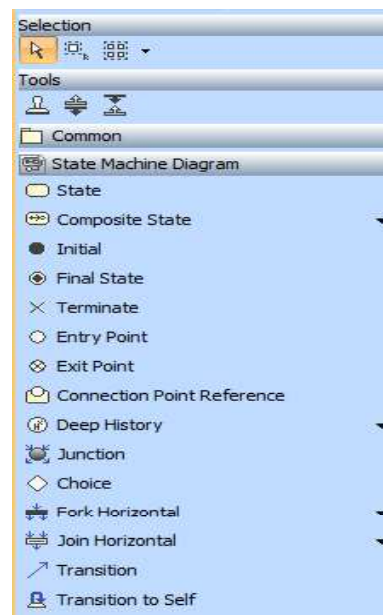


Figure B3: State Machine Diagram Toolbar

Now, the modeler should allocate the effects defined previously to each state. This is done by going to the “Relations” in the “State Specification Window” (Figure B4) and selecting “Create Incoming,” selecting “Allocate,” and selecting the appropriate effect from the allocation list shown.

There are several types of information that will be embedded into each SMD. This information is a state block representing every state on the model, the corresponding effect of the component while in that

state, the signals that the model would broadcast at every state, and the causes that would result in a state change from an operating state to a failed state, intermittent operation state, or vice versa if applicable. If all these data are not present, the findings could be unavailable or misleading.

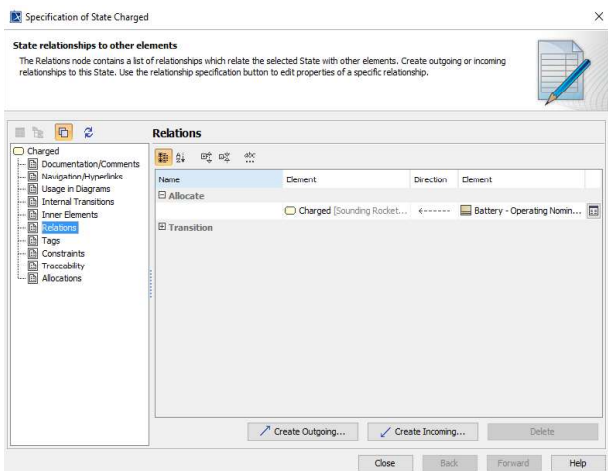


Figure B4: State Specification Window

At this point, all modeling elements have previously been defined; the only thing left for the modeler to model are the “Causes” to complete the SMD. The failure causes information should be entered into the model using “Operation” elements added underneath the component’s state machine diagram as seen in Figure B2 (e.g., four different causes were defined for the battery failure: Cell Short/Open Circuit, Joint Failure, Vibration Damage, and Component/Pin Failure). These causes are then symbolized in the state machine diagram in the form of transition lines that would trace the battery change of state from a nominal operation mode to a failed state (e.g., “Vibration damage to the battery” will result in the battery to go from the charged state to the failed state of “Wire brakeage”). While every failure could have a specific cause, as shown Figure B5, different failed states can also share a common causes.

For example in the Sounding Rocket Model, as shown in Figure B5, different states were identified for the battery, one nominal operating state of being “Charged” and four “FailedState,” including “Power harness failed short,” “Power harness failed open,” “Wire brakeage,” and “Loss of a string.”

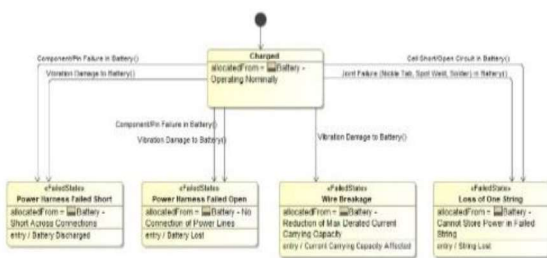


Figure B5: Battery State Machine Diagram

In the next step of modeling in MagicDraw, the modeler should to **integrate and combine individual component SMDs into the subsystem model (Step 4)**, which requires a SMD at a higher level that integrates different lower level state machine diagrams. This integration should be done in all levels of the system including parts to components, components to subsystems, subsystems to systems, and systems to system of systems (SOS).

There are two main approaches to take the model to next level: creating an entirely new state machine at a higher level, “Stand-Alone,” or using “Orthogonal” state machines. In the orthogonal SMD method, an orthogonal state machine diagram will be defined for every component and the next level effects will be addressed on that same diagram (e.g., Europa Propulsion Fuel Tank, Figure B6a).

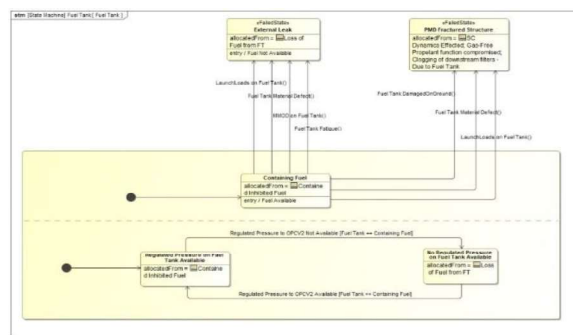


Figure B6a: Orthogonal State Machine Diagram

In the stand-alone method, a higher level diagram (in this case for Celestial ACS) is created and the transition from the nominal operation state to a failed state is defined using next level effect transitions through every component. For example, the effects of battery failure at the next higher level can be seen as “Loss of battery,” “Degraded Power Storage Capabilities,” and “Major Effect to Max Derated Current Carrying Capacity.” These three causes at the Subsystem level would be an indication of the battery being in a failed state, and thus would lead to failure of the ACS. These transition scenarios from the operating state to the failed state of the Celestial ACS are depicted in Figure B6b in the Celestial ACS State Machine Diagram.

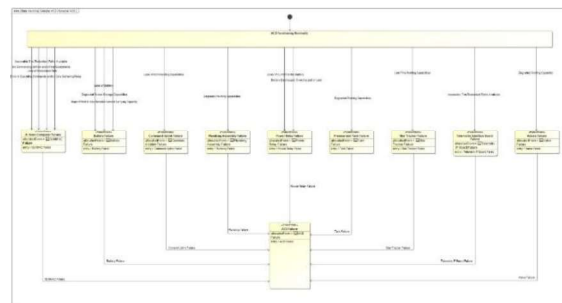


Figure B6b: Standalone State Machine Diagram

Using this method has some advantages in that it will simplify modeling parallel system elements and help clearly define next level effects using intermediate signal that will be finally shown on the FMEA outputs. On the other hand, creating a new state machine at every level will increase to total number of diagrams in each model and may make it more complicated to define a series configuration of the system elements.

The same modeling process should be repeated at every higher level until all the system levels in their entirety are integrated into a single model. Again, this process could be approached using orthogonal state machines, where at every level and subsequent component signals are used to loop back the effect of failure of a downstream/lower level component to the next level, or a standalone diagram could be created that holds all this information.

For example, in the Sounding Rocket ACS case study, a standalone SMD was used to be able to broadcast the state's signals to a higher level, and model the system level SMD to include all subsystems and corresponding failures. The Sounding Rocket system level SMD is depicted in Figure B7. As seen below, there is only one failure mode defined for every subsystem at this level for illustration purpose; thus, unlike the ACS SMD (Figure B6b), no intermediate blocks were required to carry the failures from operating to failed state.

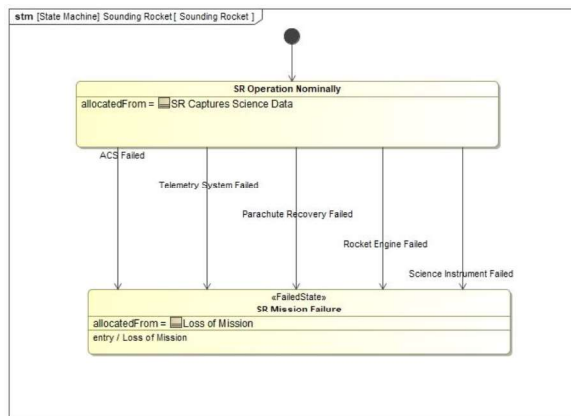



Figure B7: System level State Machine Diagram

Now that the model represents the system, the modeler should go one step further and add more detail into the model to fully support reliability analyses. These data are the “Failure Likelihood,” which is used in Quantitative FTA calculations and “Criticality Level,” which is used in FMECA Risk identifications. These parameters should be added to every effect as a value property that is used by Tietronix Plugins to make the model and output reports complete.

Further it's recommended the modeler periodically review the model content with fellow and other discipline engineers including designers, Systems Engineering, and Fault Management, as well as peer review with fellow modelers to minimize

modeling errors. Prior to generating a Failure Mode and Effect Criticality Analysis (FMECA) output using the FMECA plug-in or a Fault Tree Analysis (FTA) after modeling with the methodology described above, the modeler needs to ensure (**Verify Model (Step 5)**) with project personnel (i.e., PDL, SE, FM, CSO, RE-Lead, etc.) the following are in place and accurate:

- The behavior (Operational, Standby, and Failed) of the component is captured in state machine diagrams owned by the component.
- Operational states and potential failed states are modeled in the component-owned state machine. Failed states represent potential failure modes and are modeled as states with a “FailedState” stereotype.
- The immediate Effects of the states are modeled as blocks with “Effect” stereotypes and allocated to the corresponding states in the state machine models.
- Criticality levels are modeled as value properties of the “Effect” stereotyped blocks.
- State changes between failed and operating states are accomplished via transition lines that will carry the “cause” of each failure and/or recovery. These transitions in an orthogonal SMD (a composite state machine with at least two regions where each region can have an initial state and a final state) consist of a trigger (cause or signal) and a guard condition (a logic/conditional statement that provides the assumption of the upstream event state of failure or operation).
- Interactions between state machines are accomplished via broadcasted signal actions from one state and signal event triggers on transitions in another state machine.
- Failure likelihoods are assigned as a value property to each cause (Operation elements assigned to reach transition line) with values that represent the likelihood of the single cause only. Note: the failure rate of the component may be more than the sum of all assigned likelihoods of incoming causes (transition lines) to that failed state but should not be less than that sum and could be equal to it if all causes are modeled.

When all the modeling and assumptions are verified, then the modeler can **generate reliability reports (Step 6)**. For an FTA report (Figure B10), the user should select the “FTA” icon  (Step 6A), then select “Compute PRA” in the “List of Events” window (Figure B8) and the desired effect block as the top level event of interest to create a traditional fault tree report with Boolean quantification. One issue to note here is that, although the Boolean math used is correct,

sometimes the logic may be false (i.e., decomposing the tree into one single subsequent event). If “Compute PRA” is not selected, a qualitative Fault Tree report will be generated.

Given that the Tietronix plugin uses Failure Causes and Failure Likelihood values assigned to the causes (signal elements) to calculate probabilities, the FTA or PRA outputs are more causality probabilities, which differ from the traditionally desired scenario analyses performed at NASA/GSFC. Thus, to have a meaningful FMEA and FTA analyses using Tietronix plugins, the modeler should setup two separate SMDs in order to generate the right output.

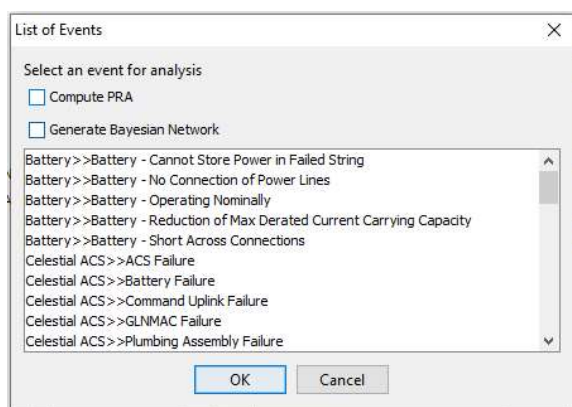


Figure B8: Effect Block List of Events

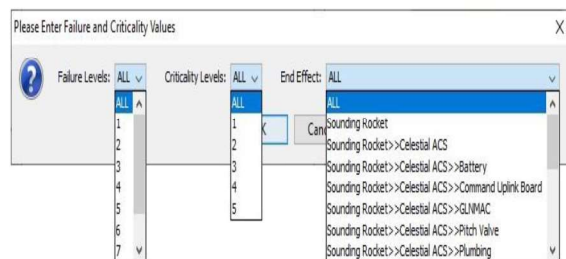


Figure B9: Effect Block List of Events


For an FMECA report (Figure B11) the user should select the “FMECA” icon  (Step 6B) and then the desired filtering by failure levels and/or criticality levels and the appropriate “End Effect” of interest from the report formatting window (Figure B9). A traditional FMECA report is generated by selecting “All” in each of the report option drop-downs. The report will include the information shown in Table B1 but will need post-generation processing by the user.

Table B1: Effect Block List of Events

Column Name	FMECA Column Description
System/Module	Populated with the top 2 “System” stereotyped blocks from the hierarchy of the unique component owning the “FailedState”
Subsystem/Sub-subsystem	Populated with the top 2 “Subsystem” stereotyped blocks from the hierarchy of the unique component owning the “FailedState”
Potential Failure Mode	Populated with the names of the states with a “FailedState” stereotype applied to it.
Immediate Failure Effect	Populated with the name of the “Effect” stereotyped block allocated to the “FailedState”
End Effect	Determined by traversing the state machine model (via broadcasted signals and signal event triggers) to the final state. This column is populated with the name of the “Effect” stereotyped block allocated to the last state traversed from the “FailedState”
Number of Independent Failures	Populated with the number of “FailedState” states encountered while traversing the path to the end effect
Other Independent Failures	Populated with the names of the other state machines and “FailedState” encountered while traversing the path from the “FailedState” to the end effect
Criticality Level	Populated with the default value assigned to the criticality Level value property of the end “Effect” block
Potential Causes	Populated with the name of the event trigger (ex: operation or signal) of transitions that terminate in the “FailedState”

In summary due to the complexity of modeling in this environment, it is highly recommend the SysML Modeling Process described above and summarized below be followed so that accurate and value-needed results are produced:

9. Create folders (a.k.a a “Package”) for every system
10. Configure /“stereotype” each block according to the input needs of the Tietronix plugin
11. Create State Machine Diagrams (SMDs)
12. Integrate and combine individual component’s SMDs into the subsystem model
13. Verify Model (iterate until verified)
14. Generate reliability reports (and perform post-generation processing).

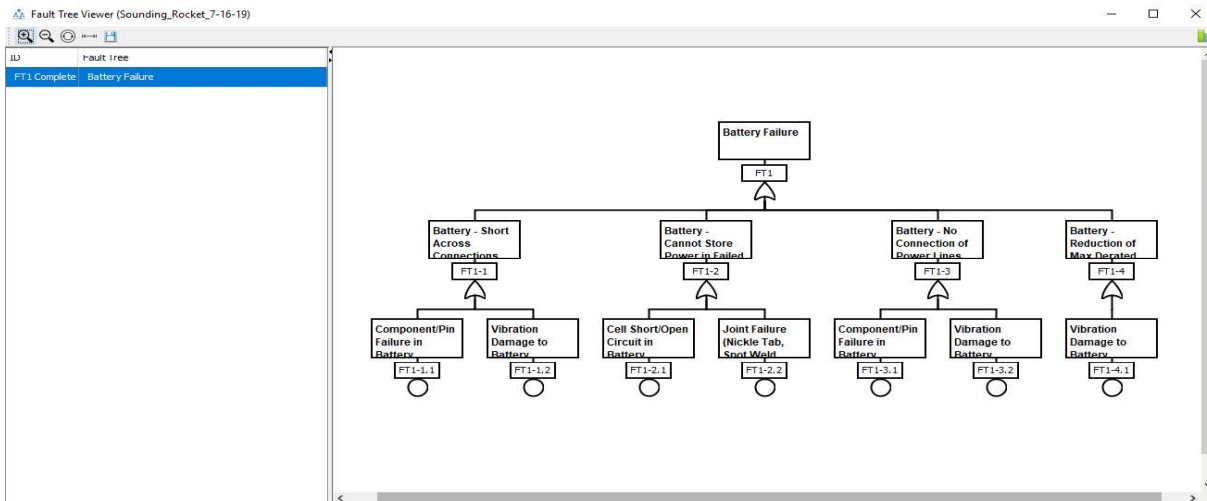


Figure B10: Sample FTA Output

Generated FMECA

System	Subsys...	Potential Failure Mode	Immediate Failure Effect	End Effect	Potential Cause(s)	Fault Propagation Path (Explict)	
Soundin...	Celestial...	Telemetry Interface Board Fail...	Telemetry IF Board Failure	No Effect	Inoperable if No Redundant Paths Available	Celestial ACS. Telemetry Interface Board Failure >> Signal: Telemetry IF Board Fail...	
Soundin...	Celestial...	Valves Failure	Valve Failure	No Effect	Degraded Pointing Capability	Celestial ACS. Valves Failure >> Signal: Valve Failure >> Celestial ACS. ACS Failure...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Battery Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Command Uplink Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Plumbing Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Power Relay Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Tank Tracker Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Star Tracker Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Telemetry IF Board Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	Valve Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	ACS Failure	ACS Failure	No Effect	GLNMAC Failure	Celestial ACS. ACS Failure >> Signal: ACS Failed >> Sounding Rocket. SR. Mission F...	
Soundin...	Celestial...	Power Harness Failed Short	Battery - Short Across Conn...	Battery - Short Across Conn...	1	Component/Pin Failure in Battery	Battery. Power Harness Failed Short >> Signal: Loss of Battery
Soundin...	Celestial...	Power Harness Failed Short	Battery - Short Across Conn...	Battery - Short Across Conn...	1	Vibration Damage to Battery	Battery. Power Harness Failed Short >> Signal: Loss of Battery
Soundin...	Celestial...	Power Harness Failed Open	Battery - No Connection of ...	Battery - No Connection of ...	1	Component/Pin Failure in Battery	Battery. Power Harness Failed Open >> Signal: Loss of Battery
Soundin...	Celestial...	Power Harness Failed Open	Battery - No Connection of ...	Battery - No Connection of ...	1	Vibration Damage to Battery	Battery. Power Harness Failed Open >> Signal: Loss of Battery
Soundin...	Celestial...	Wire Breakage	Battery - Reduction of Max ...	Battery - Reduction of Max ...	1	Vibration Damage to Battery	Battery. Wire Breakage >> Signal: Major Effect to Max Derated Current Carrying ...
Soundin...	Celestial...	Loss of One String	Battery - Cannot Store Pow...	Battery - Cannot Store Pow...	1	Joint Failure (Nickle Tab, Spot Weld, Solder) in Battery	Battery. Loss of One String >> Signal: Degraded Power Storage Capabilities
Soundin...	Celestial...	Loss of One String	Battery - Cannot Store Pow...	Battery - Cannot Store Pow...	1	Cell Short/Open Circuit in Battery	Battery. Loss of One String >> Signal: Degraded Power Storage Capabilities
Soundin...	Celestial...	Failed Short	Power Relay - Unable to Cut...	No Effect	Overstressed	Power Relay. Failed Short >> Signal: Battery Discharged, Overcharged or Lost >> ...	
Soundin...	Celestial...	Failed Open	Power Relay - Unable to Pro...	No Effect	Vibration Damage	Power Relay. Failed Open >> Signal: Loss of Current to the Battery >> Celestial A...	
Soundin...	Celestial...	Failed Open	Power Relay - Unable to Pro...	No Effect	Component/Pin Failure	Power Relay. Failed Open >> Signal: Loss of Current to the Battery >> Celestial A...	
Soundin...	Celestial...	Loss of Signal	CMD UL - Loss of Capabily	No Effect	Open Circuit on CMD UL	Command Uplink Board. Loss of Signal >> Signal: Loss of Commanding Capabily...	
Soundin...	Celestial...	Loss of Signal	CMD UL - Loss of Capabily	No Effect	Short Circuit on CMD UL	Command Uplink Board. Loss of Signal >> Signal: Loss of Commanding Capabily...	
Soundin...	Celestial...	Inoperable	Star Tracker - Degraded Att...	No Effect	Circuit Failure	Star Tracker. Inoperable >> Signal: Lost Fine Pointing Capabily >> Celestial ACS...	
Soundin...	Celestial...	Inoperable	Star Tracker - Degraded Att...	No Effect	Part Failure	Star Tracker. Inoperable >> Signal: Lost Fine Pointing Capabily >> Celestial ACS...	
Soundin...	Celestial...	Inoperable	Star Tracker - Degraded Att...	No Effect	Workmanship	Star Tracker. Inoperable >> Signal: Lost Fine Pointing Capabily >> Celestial ACS...	
Soundin...	Celestial...	Failed Close	Valve - Degraded Attitude A...	No Effect	Yaw Valve Coil Failed at Close Position	Yaw Valve. Failed Close >> Signal: Degraded Pointing Capability >> Celestial ACS. V...	
Soundin...	Celestial...	Failed Open	Valve - Loss of Pressurant	No Effect	Yaw Valve Coil Failed at Open Position	Yaw Valve. Failed Open >> Signal: Degraded Pointing Capability >> Celestial ACS. V...	
Soundin...	Celestial...	Failed Close	Valve - Degraded Attitude A...	No Effect	Pitch Valve Coil Failed at Close Position	Pitch Valve. Failed Close >> Signal: Degraded Pointing Capability >> Celestial ACS...	
Soundin...	Celestial...	Failed Open	Valve - Loss of Pressurant	No Effect	Pitch Valve Coil Failed at Open Position	Pitch Valve. Failed Open >> Signal: Degraded Pointing Capability >> Celestial ACS...	
Soundin...	Celestial...	Failed Close	Valve - Degraded Attitude A...	No Effect	Roll Valve Coil Failed at Close Position	Roll Valve. Failed Close >> Signal: Degraded Pointing Capability >> Celestial ACS. V...	

Figure B11: Sample FTA Output