Model-Based Engineering/Digital Engineering (MBE/DE) and Mission Assurance of Complex NASA Systems with Federated Modeling

Nancy J. Lindsey, Nobel Sindjui
Goddard Space Flight Center, Greenbelt, MD 20771

Key Words: Systems Engineering, Digital Engineering, Reliability, FMEA/FMECA, Risk Assessment, Fault Tree, Reliabilty Prediction, Reliability Block Diagram, Probabilistic Risk Assessment, Limited Life Items, Model-Based Mission Assurance, NASA, Goddard Space Flight Center

SUMMARY & CONCLUSIONS

The emergence of model-based engineering, with Model-Based Systems Engineering (MBSE) leading the way, is transforming design and analysis methodologies.[7] The recognized benefits to systems development include moving from document-centric information systems and document-centric project communication to a model-centric environment in which control of design changes is facilitated. In addition, a "single source of truth" that is up-to-date in all respects of the design becomes the authoritative source of data and information about the system. This promotes consistency and efficiency of the system elements as the design emerges and thereby may further optimize the design. Therefore, Mission Assurance (MA) and Systems Engineering (SE) supporting NASA missions must be integrated into model-based engineering to ensure the outputs of their solutions and analyses are relevant and value-needed to the design, development, and operational processes for failure risks assessment and communication.

Effective model-based Mission Assurance must be analyst-/modeler-agnostic while still efficiently producing complete, accurate, and more consistent artifacts (e.g., Hazard Analysis; Failure Modes, Effects, and Criticality Analysis (FMECA); Limited Life Analysis (LLA); Fault Tree Analysis (FTA); Maintainability/Availability Analysis; and Probabilistic Risk Assessment (PRA)) than traditional methods to allow engineers greater time for analysis, risk assessment, system behavior investigation (simulation), and risk-based project decision-making support. Currently, this is not fully supported by Systems Modeling Language (SysML)/MagicDraw but is best supported by the commercial tool, MADE (Maintenance Aware Design Ecosystem from PHM Technology) [11, 13]. However, systems engineering solutions in MBE/DE are currently enabled by SysML/MagicDraw and many specialized commercial tools. Therefore, to achieve a robust and unified modeling process that includes considerations from all disciplines, a method or process for model sharing and data federation must be developed that is optimized for each discipline's efforts.

In order to develop this unified modeling process, a Goddard Space Flight Center (GSFC) sponsored team has completed this federation study as part of the Model-Based Safety and Mission Assurance Initiative (MBSMAI) with the following objectives:

- Determine if it is possible (and how) to transfer/ incorporate DE/MBSE SysML/MagicDraw system elements (e.g., Bill of Material (BOM) elements) into a MADE model to create a MBSMA model and add SMA characteristics (e.g., failure concepts and functions).
- Determine if it is possible (and how) to transfer/incorporate SMA/MBSMA model findings (e.g., failure modes, fault events) into a DE/MBSE system model and emulate/attach artifacts (e.g., FMECA, Fault Tree).

To achieve these objectives, GSFC Reliability experts used previously developed SysML/MagicDraw models of mission subsystems (EUROPA Propulsion, Wallops Flight Facility (WFF) Sounding Rocket Attitude Control System (ACS), and International Space Station (ISS) Evaporator [12]) to share modeling elements with MADE for new model creation and MA analysis/support. This was successfully accomplished relatively simply by using SysML/MagicDraw tables for exporting the BOM data to Excel and using MADE import functions with NASA and MADE Pallete libraries to establish preliminary functions and failure concepts. This proved it is possible to transfer/incorporate DE/MBSE SysML/ MagicDraw system elements (e.g., BOM elements) into MADE to create a MBSMA model. In addition, each new MADE model or preliminary MBSMA model was also successfully refined with additional modeling and verified/ validated as accurate by comparing them and their SMA-artifacts to previously developed MADE models and SMA-analysis results (from MBSMAI phase 1 for the same subsystems [11, 12, 13]). This proved that comprehensive SMA characteristics (e.g., failure concepts and functions) can be added to any transfer-created MBSMA/MADE model, and accurate SMA artifacts can be generated. Further, the MA-reliability analysis results (FMECAs and Fault Trees) generated from these new models were successfully and relatively simply shared back to the corresponding SysML/MagicDraw models by using MADE's extensive Excel and pictorial output options, SysML/MagicDraw tables,

and the SysML/MagicDraw *Allocate To* attribute. This proves it is possible to transfer/incorporate SMA/MBSMA model findings into a DE/MBSE system model and emulate/attach SMA-artifacts so that MA risks can be assessed and communicated. Therefore, this study was successful at proving:

> *Model federation is easy, valid, and plausible between Mission Assurance and Engineering models if adequate modeling processes, procedures, and compatible modeling styles/structures are established and implemented.*

Note: Only transferring model-based FMECA and Fault Tree data was a study testing limitation only, other SMA/MBSMA artifacts/findings (e.g., hazards, data quality, requirement compliance, prognostics, predictions (reliability (RBDs) and availability), and maintainability analysis/assessment) that are in Excel could also be transferred with only potentially minor adjustments to the procedures (See Section 3 and Appendix A) and modeling constructs used in this study.

Consequently, this study recommends the following:

1) Modify and expand the modelling processes and controls recommended in previous studies [11, 12, 13] as follows: 1) Establish a multi-discipline modeling team (Systems Engineering (SE) and Safety and Mission Assurance (SMA) at a minimum); 2) Establish modeling responsibilities (e.g., SE's model requirements and system configuration hierarchy in Block Definition Diagrams (BDDs), Reliability Engineers (REs) model failure behaviors/characteristics/controls and codify with input from the design and systems teams, configuration element-functionality and interface/flow details get included in Functional Block/Wire Diagrams, Safety Engineers model hazards); 3) Complete SMA-modeling, based on importing/ receiving modelling elements and data from the system model, by (3A) creating a BOM in the system model and exporting that to Excel, (3B) importing the exported BOM into the SMA-model, and (3C) performing SMA modeling; 4) Produce SMA artifacts and share resulting data between modelling elements by (4A) exporting SMA artifacts and artifact-data items in Excel and (4B) ingesting/emulating those artifacts in the system model in tables and attachments; 5) Validate and refine modelling (and designs) until a final and acceptable result is achieved; and 6) Share modeling with future missions.

2) Include a hierarchical system decomposition in System SysML/ MagicDraw models in the BDD with consistent ownership relationships and application of federation stereotypes (e.g., *system, subsystem, unit, piece part*) so a BOM table for federation and other purpose can be created. This BDD can also be used for requirement allocation and work breakdown structuring.

3) Specify, during the establishment of roles and responsibilities, that SMA will model/codify, with input from the design and systems teams, the additional functional details to conduct SMA analysis/assessments. This is more than in traditional block or interface diagrams would provide (e.g., Propulsion Latch Valves would need power/command inputs and telemetry outputs as well as material flows defined to fully characterize their functionality) and would elevate understanding for development teams as well. These characterizations can be shared back to the system SysML/MagicDraw model as a graphic now (as is the current/ traditional project communications method). Soon they will be able to be ingested by SysML/MagicDraw as an Interface Block Diagram (IBD) with the completion of MADE's functional diagram to IBD transfer capability development.

4) Standardize the use of clear and differentiable naming of elements in all models (no two modeling elements, packages, behaviors, or state machines should share the same name) and libraries should be used for commonly used architectures or elements to avoid having cross-referencing and allocation issues.

5) Use the data federation procedure (Appendix A) developed during this study, the federation supporting profiles/stereotypes shown Appendix B, and MADE to complete SMA analysis/ assessment integration with the system SysML/MagicDraw model. The use of MADE is recommended since MADE can support SMA, has been evaluated to have the best model-based SMA Return on Investment (ROI) based on earlier study phases [12, 13], and can produce Excel reports/exports of hazard, data quality, requirement compliance, prognostics, reliability (FMECA and Fault Trees), predictions (availability and reliability (RBDs)), and maintainability analysis/assessment.

6) Use the portion of the data federation procedure for transferring/incorporating SMA/MBSMA model findings (e.g., failure modes, fault events) (Appendix A) and the federation supporting profiles/stereotypes (Appendix B), developed during this study, for any SMA analyses or assessments (model-based or not) that can be output/delivered in Excel. But great care and potential re-processing to match modeled configurations and item names should be used.

7) Continue the currently planned additional federation and model-synchronization testing, optimizations, and developments to:
   - Determine a method for sharing MADE functions and flows and configuration updates/decompositions with SysML/ MagicDraw to update MBSE model's containment tree (structure and content) and BDD; and create/replace IBD.
   - Determine a method for sharing SysML/MagicDraw configuration updates/decompositions with MADE without updating the entire model.
   - Identify and test additional large table ingestion methods in SysML/MagicDraw to avoid excessive wait times.
   - Identify and resolve any federation issues at the mission-level, including but not limited to the potential for system models to grow quite large and slow SysML/MagicDraw performance.
   - Test the federation of data from other model-based SMA-discipline outputs that have been tested (and non-model SMA data imports) to verify assumed capabilities.
   - Expand MBSMA to include Software Assurance and Quality Engineering.

In summary, the provided results, findings, recommendations, and guidance can be used now by NASA/GSFC to advance digital transformation efforts and to assist in the establishment of SMA-to-SE and SE-to-SMA modeling transitions, data transfers, and collaborations. However, they were formulated based on subsystem-model testing only, therefore this study team has developed a path forward to conduct the needed testing to ensure that the solutions and procedures of this study are scalable to mission-level modeling/assurance-efforts and other SMA-disciplines. As a result, GSFC and the MBSMAI team plan to execute a FY25 IRAD to perform mission-level federation testing, expand/test the feasibility of SMA-modeling including Software Assurance (SWA) efforts (Quality Engineering testing is planned for FY26), continue maintenance of the SMA-reference model to jumpstart new modeling, and continue sharing modeling tips/guidance broadly.

1 BACKGROUND

"Mission Assurance is a full life-cycle engineering process to identify and mitigate design, production, test, and field support deficiencies threatening mission success." [9]

1.1 Mission Assurance Engineering

Mission assurance engineering at NASA/GSFC is a framework of methodologies, analyses, tools, and processes that ensure each system's safety; reliability, maintainability, and availability (RMA); quality; and software robustness via risk assessments and analyses to assess and manage mission "lifetime" engineering risks and impacts of hazards/failures, mitigations, corrective actions, and recovery strategies.[9, 10] As a result, each SMA discipline is an integral part of NASA/GSFC's Continuous Risk Management (CRM) (See Figure 1). In continuous risk management risks are identified and analyzed/researched, then a plan is developed to handle (e.g., mitigate, watch, accept, or escalate) the risks, and ultimately the risks are monitored for realization and/or modification.

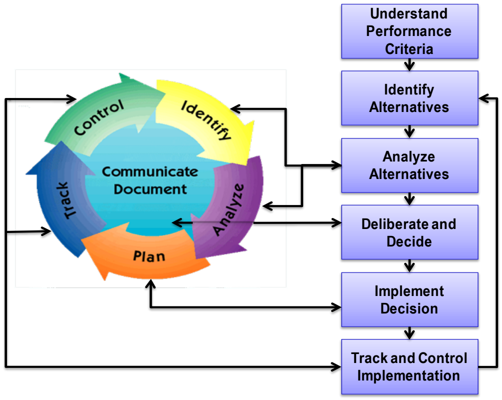


Figure 1- RIDM-CRM Risk Management Process Flow [1]

1.1 Modeling Tools Utilized

1.1.1 *MADE*

The modeling tool Maintenance Aware Design ecosystem (MADE) provides a suite of software tools that can be used to design, assess, and optimize Prognostics and Health Management systems for use in a wide variety of high-risk industries where safety and reliability are critical, using model-based engineering techniques. The MADE modelling environment, shown in Figure 2, provides specific analysis workflows for reliability, including Reliability Allocation, Reliability Block Diagrams, Markov Analysis, and Reliability/Availability Analysis with multiple failure distribution methodologies to produce and validate the reliability requirements for a system at each stage of the design process. These analyses allow for on-demand generation of FMEA, FMECA, and Common Mode Analysis and Functional Fault Tree Analysis reports.

A MADE model of a system uses a graphical Functional Block Diagram and automates the propagation of functional failures in a system to establish syndromes or signatures of failure based on the underlying physics of failure. This information is used to generate and optimize diagnostic sensor placement based on the probability of detection (PoD) of potential failures. Additionally, the propagation and sensor information can be utilized to ensure Fault Detection and Isolation and life/maintenance/diagnostic rule implementation is balanced with cost, weight, and risks and to produce diagnostic rules based on the selected combination of sensors.

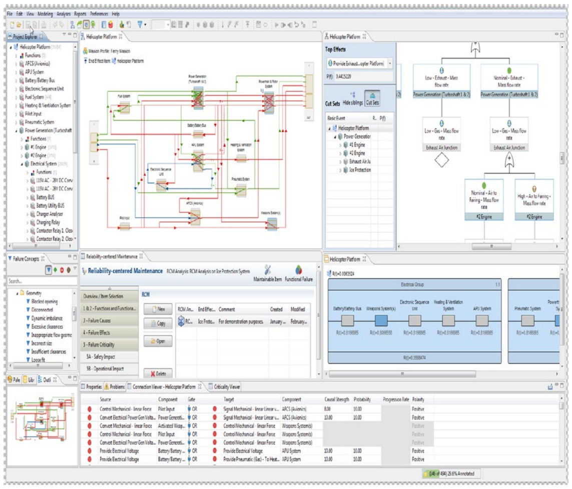


Figure 2 – MADE Modeling Environment [3]

1.1.2 *SysML/MagicDraw*

This study's Systems Modeling Language (SysML/Magic Draw) modeling tool (v19.0) was an extension of UML 2.0, with Tietronix plug-ins (FaultTree (18.0), FMECA (18.0), MBSE Plugin (18.0), Methodology Wizards (19.0SP2), Model Obfuscator (19.0), and Product Line Engineering (19.0SP2)), designed to support modeling for System Engineering and Reliability. It is a general-purpose graphical modeling language for analyzing, designing, and verifying complex systems, which may include hardware, software, information, personnel, procedures, and facilities. A SysML system model consists of a Functional/Behavioral Model, Performance Model, Structure/Component Model, and Other Engineering Analysis Models (Figure 3) to integrate system requirements with engineering disciplines. In order to perform SMA analyses, the SysML plugins of FMECA, FTA, and PRA (developed by NoMagic, CAMEO, Tietronix, and modeling teams) must be executed against the SysML system model.

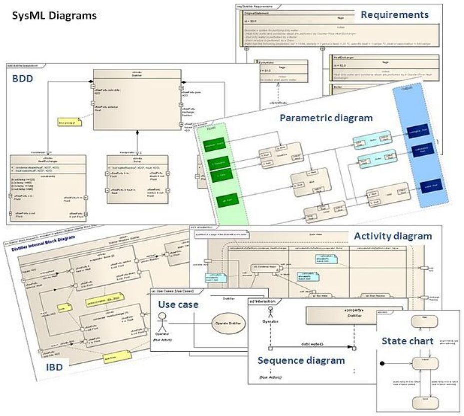


Figure 3: SysML Diagrams [2]

2 *TEST METHODOLOGY*

The test methodology used to assess MBSMA and MBE/DE model federation assumed that a multidisciplinary modeling team had been established for each test case and the responsibility for establishing each system's constituent elements and overall risk management had been allocated to the Systems Engineering and Design teams, and those efforts would be performed in SYSML/MagicDraw, while the codification of functional/interface details (e.g., dependencies, flows, functions) and assurance/failure analysis had been allocated to the SMA team, and those would be done in MADE. Therefore, this study 1) used previously developed models of mission subsystems (EUROPA Propulsion, Wallops Flight Facility (WFF) Sounding Rocket Attitude Control System (ACS), and International Space Station (ISS) Evaporator [12]) in SysML/MagicDraw to generate system Bill of Materials (BOMs) to share modeling elements with MADE for new model creation; 2) ingested the exported modeling elements into MADE and added the failure concepts and functions/flows not defined SYSML/MagicDraw to complete models for MA analysis/support; 3) compared resultant MADE models to previously manually-developed MADE models [phase1] to ensure modeling accuracy; 4) generated and compared representative SMA and analysis results (e.g., FMECAs and FTAs [phase1]) for the same subsystems to ensure modeling and analysis accuracy was achieved; 5) shared SMA modeling results (MA-reliability data) back to SysML/MagicDraw models so that MA risks can be assessed and communicated. Each of these processes was tested on all test cases and evaluated for accuracy, consistency, and efficiency to attain optimal federated modeling processes and best practices, and to identify modeling environment necessities for future use.

2.1 *EUROPA Propulsion (Mechanical) Test Case*

The EUROPA propulsion subsystem, shown in Figure 4, provided by GSFC, will be used on a Europa flyby mission to the Jupiter system to perform repeated close flybys of the giant planet's large moon Europa to investigate its potential habitability. The mission will collect information on Europa's ice shell thickness, composition, and surface geomorphology.

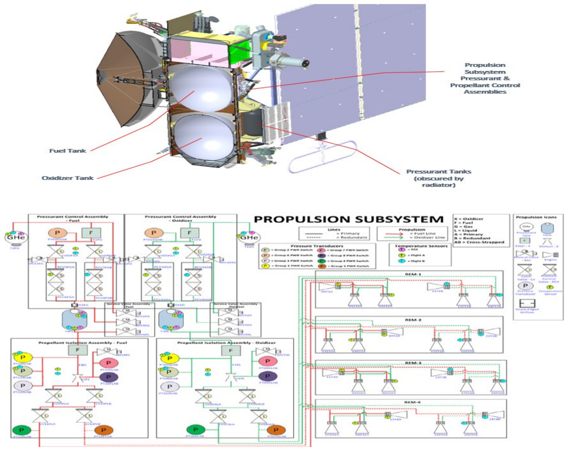


Figure 4 – EUROPA Propulsion System [8]

2.2 *Sounding Rocket (Electronic)Test Case*

The sounding rocket test case models the Celestial ACS (CACS) Subsystem of a Wallops Flight Facility Sounding Rocket as defined in the Wallops Sounding Rocket Handbook [4]. A sounding rocket carries experiments to altitudes between 50 and 1,500 km and flies nearly parabolic trajectories while its Celestial ACS is used to align sounding rocket payloads towards celestial targets. This attitude control subsystem is used for flights investigating targets that can either be acquired and tracked with a star tracker or pointed at by using nearby celestial targets as a reference. The subsystem is composed of the elements shown in the Figure 5.

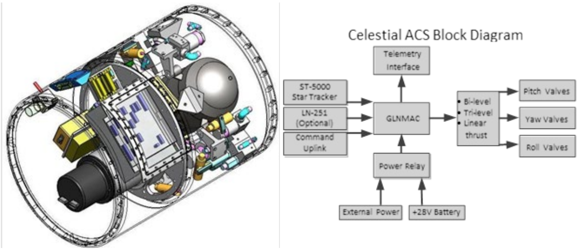


Figure 5 – Sounding Rocket Subsystem [4,11]

2.3 *ISS-Evaporator (Electromechanical)Test Case*

The ISS-Evaporator test case used a JSC design solution for ISS brine evaporation (CapiBRIC - Capillary-Based Brine Residual In-Containment) [5, 6]. The CapiBRIC system uses unique containment geometry, capillary flow, and static phase separation to enable water evaporation in a microgravity environment. CapiBRIC contains a capillary drying unit within a drying chamber (See Figure 6). This design allows water to be recovered from the clean water vapor evaporating from the free surfaces while leaving waste brine solids behind. In this way CapiBRIC is designed to help mitigate limitations of the current ISS water recovery system that cause unfeasible water storage issues for long duration space missions.

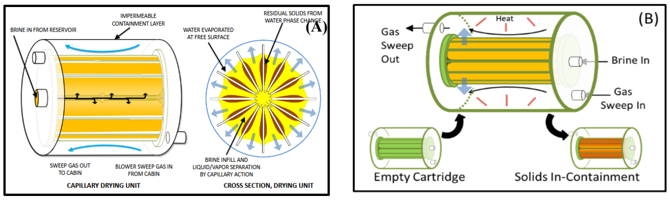


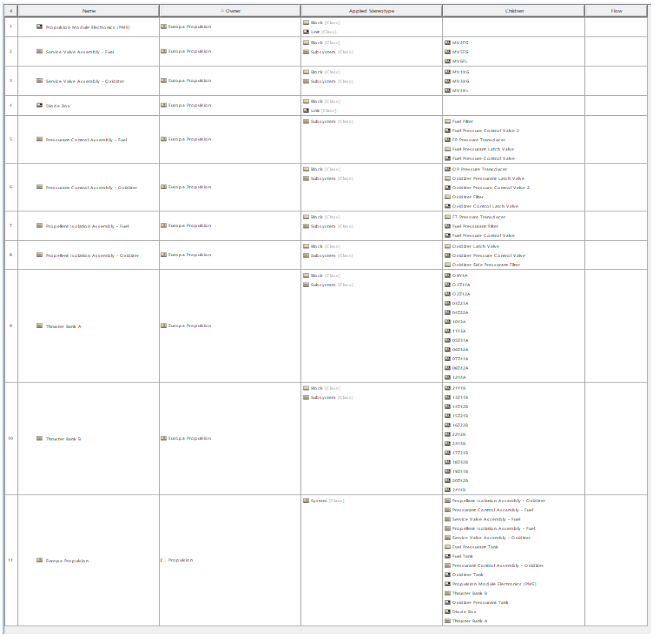
Figure 6 – ISS- Evaporator System (CapiBRIC) [5, 6]

3 *TEST PROCESS RESULTS AND LESSONS LEARNED*

3.1 *Generate and Export Bill of Materials (BOMs)*

Using the process shown in Appendix A (steps I.A – I.D), the previously developed SysML/MagicDraw models were used to generate, in generic SysML/MagicDraw tables, BOMs, as shown in Figures 7 - 9. Therefore, each legacy SysML/MagicDraw model, from phase 1 of MBSMAI, was modified to ensure that each item had the stereotypes of *system, subsystem, unit,* or *piece part* applied as appropriate from the existing or federation-created stereotypes/profiles (See Appendix B) to correct/modify the owner relationship to match the configuration decomposition shown in the IDB/BDD or system documentation (See Appendix A: steps I.A.1 – I.B.2 updates were reflected in the diagram and containment tree of each model); and a generic BOM table was added and exported to Excel for importing (See Appendix A: steps I.C.1 – I.C.2, I.D). While it is essential that this process be completed for SMA model integration/federation, it can be completed either by design/systems or SMA modelers, and it is most effective and efficient if it is completed prior to SMA modeling (best/recommended option). If parallel modeling efforts are used for this process (ineffective/inefficient option), they will require significant collaboration and will likely require repeated rework on both sides and only save the BOM creation/exporting efforts. If the decision is made to forgo this BOM creation process, then design element sharing will be impossible or manual only, models will need to be kept in sync manually (which is not recommended), and it will again only save the BOM creation/exporting efforts since stereotyping of each design element in the model (which is most rapidly done in a model's IDB/BDD) will still be needed for failure data importing from SMA-modeling and cannot be avoided, or modeling federation with SMA-modeling/findings will be effectively lost entirely.

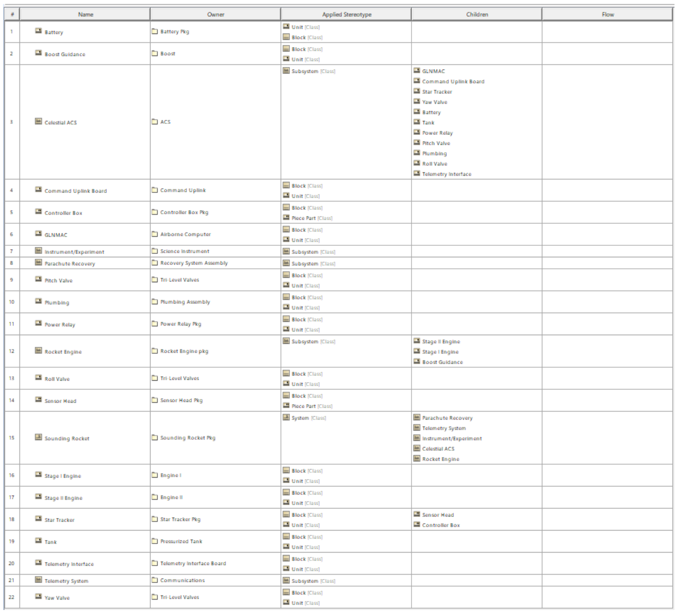
To complete this process for the EUROPA propulsion test case, the legacy EUROPA propulsion SysML/MagicDraw model [12] was modified to apply *subsystem, unit,* or *piece part* stereotypes to most items and to create a BDD with owner relationship to match the configuration decomposition shown in the containment tree since only a partial one existed at the study start. This was a relatively simple process since the legacy model had a straight-forward structure in the containment tree. When the generic BOM table was added to the model, it resulted in the identification of 61 items, 11 being parents/owners (or potential parents/owners) and 50 being children/owned items (See Figure 7).



#	Name	Owner	Applied Stereotype	Children	Flow
1	Propulsion Module Electronics (PME)	Europa Propulsion	Block [Class] Unit [Class]		
2	Service Valve Assembly - Fuel	Europa Propulsion	Block [Class] Subsystem [Class]	MV2FG MV5FG MV6FL	
3	Service Valve Assembly - Oxidizer	Europa Propulsion	Block [Class] Subsystem [Class]	MV1XG MV3XG MV4XL	
4	Diode Box	Europa Propulsion	Block [Class] Unit [Class]		
5	Pressurant Control Assembly - Fuel	Europa Propulsion	Subsystem [Class]	Fuel Filter Fuel Pressure Control Valve 2 FP Pressure Transducer Fuel Pressurant Latch Valve Fuel Pressure Control Valve	
6	Pressurant Control Assembly - Oxidizer	Europa Propulsion	Block [Class] Subsystem [Class]	OP Pressure Transducer Oxidizer Pressurant Latch Valve Oxidizer Pressure Control Valve 2 Oxidizer Filter Oxidizer Control Latch Valve	
7	Propellent Isolation Assembly - Fuel	Europa Propulsion	Block [Class] Subsystem [Class]	FT Pressure Transducer Fuel Pressurant Filter Fuel Pressure Control Valve	
8	Propellent Isolation Assembly - Oxidizer	Europa Propulsion	Block [Class] Subsystem [Class]	Oxidizer Latch Valve Oxidizer Pressure Control Valve Oxidizer Side Pressurant Filter	
9	Thruster Bank A	Europa Propulsion	Block [Class] Subsystem [Class]	09Y1A 01Z11A 02Z12A 03Z21A 04Z22A 10Y2A 11Y3A 05Z31A 06Z32A 07Z41A 08Z42A 12Y4A	
10	Thruster Bank B	Europa Propulsion	Block [Class] Subsystem [Class]	21Y1B 13Z11B 14Z12B 15Z21B 16Z22B 22Y2B 23Y3B 17Z31B 18Z32B 19Z41B 20Z42B 24Y4B	
11	Europa Propulsion	Propulsion	System [Class]	Propellent Isolation Assembly - Oxidizer Pressurant Control Assembly - Fuel Service Valve Assembly - Fuel Propellent Isolation Assembly - Fuel Service Valve Assembly - Oxidizer Fuel Pressurant Tank Fuel Tank Pressurant Control Assembly - Oxidizer Oxidizer Tank Propulsion Module Electronics (PME) Thruster Bank B Oxidizer Pressurant Tank Diode Box Thruster Bank A	

Figure 7. EUROPA BOM in SYSML/MagicDraw

For the Sounding Rocket test case, the legacy Sounding Rocket SysML/MagicDraw model [12], 16 items were modified to apply *unit* or *piece part* stereotypes, and several ownership relationships were corrected/modified to match the configuration decomposition shown in the BDD and reference documentation [4]. This was a very simple process since the legacy model had a straight-forward structure and a complete BDD diagram. Then, when the generic BOM table was added to this model, it resulted in the identification of 43 items, 22 being parents/owners (or potential parents/owners) and 21 being children/owned (See Figure 8).

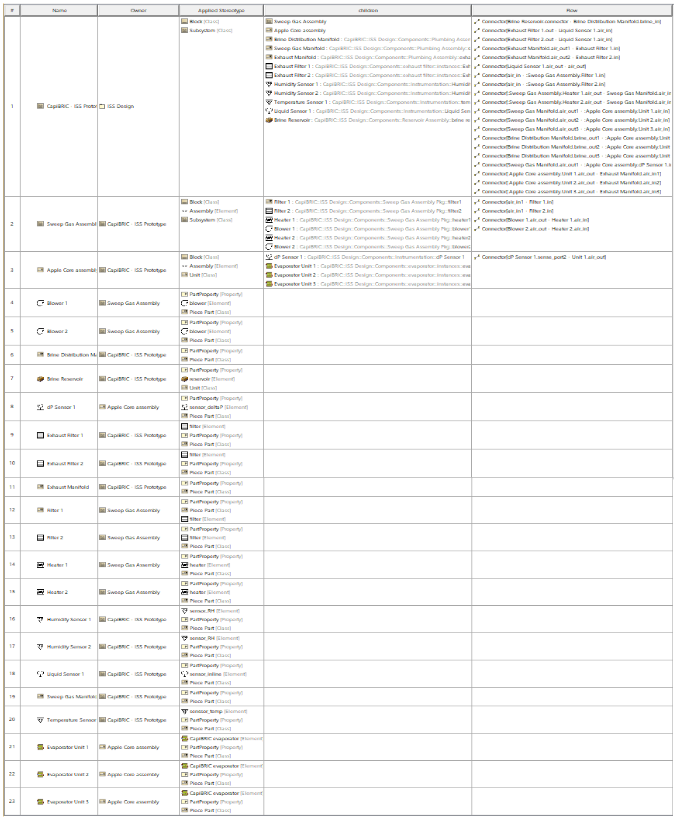


#	Name	Owner	Applied Stereotype	Children	Flow
1	Battery	Battery Pkg	Unit [Class] Block [Class]		
2	Boost Guidance	Boost	Block [Class] Unit [Class]		
3	Celestial ACS	ACS	Subsystem [Class]	GLNMAC Command Uplink Board Star Tracker Yaw Valve Battery Tank Power Relay Pitch Valve Plumbing Roll Valve Telemetry Interface	
4	Command Uplink Board	Command Uplink	Block [Class] Unit [Class]		
5	Controller Box	Controller Box Pkg	Block [Class] Piece Part [Class]		
6	GLNMAC	Airborne Computer	Block [Class] Unit [Class]		
7	Instrument/Experiment	Science Instrument	Subsystem [Class]		
8	Parachute Recovery	Recovery System Assembly	Subsystem [Class]		
9	Pitch Valve	Tri-Level Valves	Block [Class] Unit [Class]		
10	Plumbing	Plumbing Assembly	Block [Class] Unit [Class]		
11	Power Relay	Power Relay Pkg	Block [Class] Unit [Class]		
12	Rocket Engine	Rocket Engine pkg	Subsystem [Class]	Stage II Engine Stage I Engine Boost Guidance	
13	Roll Valve	Tri-Level Valves	Block [Class] Unit [Class]		
14	Sensor Head	Sensor Head Pkg	Block [Class] Piece Part [Class]		
15	Sounding Rocket	Sounding Rocket Pkg	System [Class]	Parachute Recovery Telemetry System Instrument/Experiment Celestial ACS Rocket Engine	
16	Stage I Engine	Engine I	Block [Class] Unit [Class]		
17	Stage II Engine	Engine II	Block [Class] Unit [Class]		
18	Star Tracker	Star Tracker Pkg	Block [Class] Unit [Class]	Sensor Head Controller Box	
19	Tank	Pressurized Tank	Block [Class] Unit [Class]		
20	Telemetry Interface	Telemetry Interface Board	Block [Class] Unit [Class]		
21	Telemetry System	Communications	Subsystem [Class]		
22	Yaw Valve	Tri-Level Valves	Block [Class] Unit [Class]		

Figure 8. Sounding Rocket BOM in SysML/MagicDraw

For the CapiBRIC test case, the legacy CapiBRIC SysML/MagicDraw model [12], 17 items were modified to apply *subsystem, unit,* or *piece part* stereotypes, and 3 ownership relationships were created to match the configuration decomposition shown in the International Space Station (ISS) CapiBRIC instantiation IBD and CapiBRIC documentation [5, 6]. This was a challenging process since the legacy model not only included the ISS instantiation, but also a reference architecture of CapiBRIC elements to develop each specific instantiation from that was not easily differentiated from the ISS-instantiations. This meant that each stereotype was not simply applied to an IBD item but was also applied to the referenced *item type,* once that was located in the containment tree. However, since the reference architecture used the exact same names as the ISS-instantiation, this caused multiple updating iterations to get correct. Therefore, it is recommended clear and differentiable naming be employed if libraries of components are not available and the reference items will co-exist (not recommended) in a model with the specific instantiation to avoid federation or other issues (e.g., requirement allocation).

Once these stereotypes and ownership-settings were completed, then the generic BOM table could be added to this model, which resulted in the identification of 23 items, 3 being parents/owners (or potential parents/owners) and 20 being children/owned, and 25 flows being captured (See Figure 9). The captured flows described interfaces between items (e.g., connectors) but did not have sufficient interface characteristics for interface detailed modeling or failure/propagation analysis, so they were used only to verify ownership settings in this study.



#	Name	Owner	Applied Stereotype	children	Flow
1	CapiBRIC · ISS Protot	ISS Design	Block [Class] Subsystem [Class]	Sweep Gas Assembly Apple Core assembly Brine Distribution Manifold : CapiBRIC::ISS Design::Components::Plumbing Assem Sweep Gas Manifold : CapiBRIC::ISS Design::Components::Plumbing Assembly::s Exhaust Manifold : CapiBRIC::ISS Design::Components::Plumbing Assembly::exha Exhaust Filter 1 : CapiBRIC::ISS Design::Components::exhaust filter::instances::Exh Exhaust Filter 2 : CapiBRIC::ISS Design::Components::exhaust filter::instances::Exh Humidity Sensor 1 : CapiBRIC::ISS Design::Components::Instrumentation::Humidit Humidity Sensor 2 : CapiBRIC::ISS Design::Components::Instrumentation::Humidit Temperature Sensor 1 : CapiBRIC::ISS Design::Components::Instrumentation::tem Liquid Sensor 1 : CapiBRIC::ISS Design::Components::Instrumentation::Liquid Sen Brine Reservoir : CapiBRIC::ISS Design::Components::Reservoir Assembly::brine re	Connector[Brine Reservoir.connector - Brine Distribution Manifold.brine_in] Connector[Exhaust Filter 1.out - Liquid Sensor 1.air_in] Connector[Exhaust Filter 2.out - Liquid Sensor 1.air_in] Connector[Exhaust Manifold.air_out1 - Exhaust Filter 1.in] Connector[Exhaust Manifold.air_out2 - Exhaust Filter 2.in] Connector[Liquid Sensor 1.air_out - air_out] Connector[air_in - :Sweep Gas Assembly.Filter 1.in] Connector[air_in - :Sweep Gas Assembly.Filter 2.in] Connector[:Sweep Gas Assembly.Heater 1.air_out - Sweep Gas Manifold.air_in Connector[:Sweep Gas Assembly.Heater 2.air_out - Sweep Gas Manifold.air_in Connector[Sweep Gas Manifold.air_out1 - :Apple Core assembly.Unit 1.air_in] Connector[Sweep Gas Manifold.air_out2 - :Apple Core assembly.Unit 2.air_in] Connector[Sweep Gas Manifold.air_out3 - :Apple Core assembly.Unit 3.air_in] Connector[Brine Distribution Manifold.brine_out1 - :Apple Core assembly.Unit Connector[Brine Distribution Manifold.brine_out2 - :Apple Core assembly.Unit Connector[Brine Distribution Manifold.brine_out3 - :Apple Core assembly.Unit Connector[Sweep Gas Manifold.air_out1 - :Apple Core assembly.dP Sensor 1.i Connector[:Apple Core assembly.Unit 1.air_out - Exhaust Manifold.air_in1] Connector[:Apple Core assembly.Unit 2.air_out - Exhaust Manifold.air_in2] Connector[:Apple Core assembly.Unit 3.air_out - Exhaust Manifold.air_in3]
2	Sweep Gas Assembl	CapiBRIC · ISS Prototype	Block [Class] Assembly [Element] Subsystem [Class]	Filter 1 : CapiBRIC::ISS Design::Components::Sweep Gas Assembly Pkg::filter1 Filter 2 : CapiBRIC::ISS Design::Components::Sweep Gas Assembly Pkg::filter2 Heater 1 : CapiBRIC::ISS Design::Components::Sweep Gas Assembly Pkg::heater1 Blower 1 : CapiBRIC::ISS Design::Components::Sweep Gas Assembly Pkg::blower1 Heater 2 : CapiBRIC::ISS Design::Components::Sweep Gas Assembly Pkg::heater2 Blower 2 : CapiBRIC::ISS Design::Components::Sweep Gas Assembly Pkg::blower2	Connector[air_in1 - Filter 1.in] Connector[air_in1 - Filter 2.in] Connector[Blower 1.air_out - Heater 1.air_in] Connector[Blower 2.air_out - Heater 2.air_in]
3	Apple Core assembl	CapiBRIC · ISS Prototype	Block [Class] Assembly [Element] Unit [Class]	dP Sensor 1 : CapiBRIC::ISS Design::Components::Instrumentation::dP Sensor 1 Evaporator Unit 1 : CapiBRIC::ISS Design::Components::evaporator::instances::eva Evaporator Unit 2 : CapiBRIC::ISS Design::Components::evaporator::instances::eva Evaporator Unit 3 : CapiBRIC::ISS Design::Components::evaporator::instances::eva	Connector[dP Sensor 1.sense_port2 - Unit 1.air_out]
4	Blower 1	Sweep Gas Assembly	PartProperty [Property] blower [Element] Piece Part [Class]		
5	Blower 2	Sweep Gas Assembly	PartProperty [Property] blower [Element] Piece Part [Class]		
6	Brine Distribution Ma	CapiBRIC · ISS Prototype	PartProperty [Property] Piece Part [Class]		
7	Brine Reservoir	CapiBRIC · ISS Prototype	PartProperty [Property] reservoir [Element] Unit [Class]		
8	dP Sensor 1	Apple Core assembly	PartProperty [Property] sensor_deltaP [Element] Piece Part [Class]		
9	Exhaust Filter 1	CapiBRIC · ISS Prototype	filter [Element] PartProperty [Property] Piece Part [Class]		
10	Exhaust Filter 2	CapiBRIC · ISS Prototype	filter [Element] PartProperty [Property] Piece Part [Class]		
11	Exhaust Manifold	CapiBRIC · ISS Prototype	PartProperty [Property] Piece Part [Class]		
12	Filter 1	Sweep Gas Assembly	PartProperty [Property] Piece Part [Class] filter [Element]		
13	Filter 2	Sweep Gas Assembly	PartProperty [Property] filter [Element] Piece Part [Class]		
14	Heater 1	Sweep Gas Assembly	PartProperty [Property] heater [Element] Piece Part [Class]		
15	Heater 2	Sweep Gas Assembly	PartProperty [Property] heater [Element] Piece Part [Class]		
16	Humidity Sensor 1	CapiBRIC · ISS Prototype	sensor_RH [Element] PartProperty [Property] Piece Part [Class]		
17	Humidity Sensor 2	CapiBRIC · ISS Prototype	sensor_RH [Element] PartProperty [Property] Piece Part [Class]		
18	Liquid Sensor 1	CapiBRIC · ISS Prototype	PartProperty [Property] sensor_inline [Element] Piece Part [Class]		
19	Sweep Gas Manifolc	CapiBRIC · ISS Prototype	PartProperty [Property] Piece Part [Class]		
20	Temperature Sensor	CapiBRIC · ISS Prototype	senssor_temp [Element] PartProperty [Property] Piece Part [Class]		
21	Evaporator Unit 1	Apple Core assembly	CapiBRIC evaporator [Element] PartProperty [Property] Piece Part [Class]		
22	Evaporator Unit 2	Apple Core assembly	CapiBRIC evaporator [Element] PartProperty [Property] Piece Part [Class]		
23	Evaporator Unit 3	Apple Core assembly	CapiBRIC evaporator [Element] PartProperty [Property] Piece Part [Class]		

3.2 *Ingest Modeling Elements and Finalize SMA Modeling*

To ensure that accurate, consistent, and efficient SMA analysis is possible, a system's BOM must be imported into MADE effectively, then modeling must be completed, and that modeling must be verified/validated to assure that applicable and accurate SMA results can be provided to the project and back to the relevant SysML/MagicDraw model.

3.2.1 Ingest Modeling Elements and Model in MADE

To import the exported SysML/MagicDraw BOMs into MADE, the Excel file must be post-processed to create an entry for each parent-child relationship (even if there is no child) from the exported parent-children data using the procedure shown in Appendix A (steps II.A.1.1 - II.A.1.3). This allows the MADE import feature to create modeling elements with the appropriate decomposition and relationships. In addition, during the MADE import process, shown in Appendix A (steps II.A.2.1 - II.A.2.6), each imported element must be matched with a matching element from the draft NASA and MADE Pallete libraries to establish preliminary functions and failure concepts (that can be modified/customized later), so the resultant model is complete when importing is finished. For this study, that meant a draft NASA MADE library of 23 components with functions and failure diagrams was created, which can and should be used for future modeling efforts. Note: None of the legacy SysML/MagicDraw models had functions and flows that were sufficient for full system characterization (e.g., functionality and inter-connectivity, failure analysis, and failure propagation), so the importing of these was not studied. But there may be the ability to pass this level of detail back to SysML/MagicDraw from MADE in the near future.

For the Europa propulsion test case, MADE ingestion resulted in a MADE Europa Propulsion model with 16 main elements, 10 of which have underlying system decomposition. The preliminary functions and failure concepts imported with each modeling element were then refined or added to manually at every level (by following steps 4-6 and 8 in *MADE Modeling Techniques & Tips* [14] (as shown in Appendix II.B)) as needed to complete the model and represent the functions, flows, and other parameters to represent the Europa propulsion system (see Figure 10) and its underlying systems' functionality and interconnectivity. During this model's completion process, it was noted that the SysML/MagicDraw thruster bank and propellant isolation assembly (PIA) for fuel did not match Europa documentation [8], therefore the 2 imported thruster banks were changed to 4 REMs, and a latch valve was added to the PIA for fuel. The result is that each of the model's 64 elements can be used to assess system failure risks accurately, but the MADE and SysML/MagicDraw models are slightly out of synchronization (effects of this will be discussed in section 3.3).

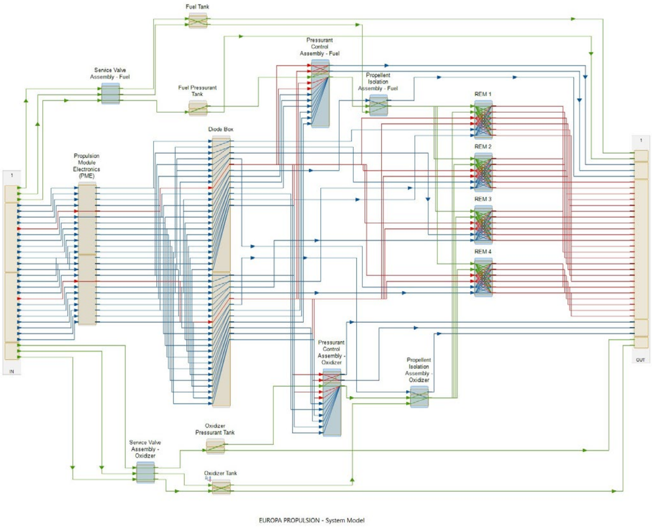


Figure 10. Europa Propulsion MADE Model

For the Sounding Rocket test case, MADE ingestion resulted in a MADE Sounding Rocket model with 5 elements, 2 of which have multiple layers of underlying system decomposition. Specifically, the celestial ACS element (the focus of the phase 1 study [11]) consists of 12 components, 1 of which is even further decomposed. Again, the preliminary functions and failure concepts imported with each modeling element were then refined or added to manually at every level (by following the steps 4-6 and 8 in *MADE Modeling Techniques & Tips* [14] (as shown in Appendix II.B)) as needed to complete the model and represent the functions, flows, and other parameters to represent the sounding rocket system (See Figure 11) and its underlying systems' functionality and interconnectivity. During this model's completion process, additional modeling of sounding rocket systems not modeled during MBSMAI phase 1 was completed [11]. This meant that an underlying model of the rocket engine system was created. This includes two engines and a boost guidance system that has 4 underlying systems (i.e., IMU, computer, stabilizer, and wing) of its own. In addition, the celestial ACS's missing IMU, per sounding rocket documentation [8], was added. Further, while the SysML/MagicDraw model did not decompose the star tracker in its celestial ACS model, the NASA library does decompose this element; so, when ingested, this decomposition became present in the MADE sounding rocket model. But for failure reporting consistency with the SysML/MagicDraw model, the underlying modeling was left un-connected/incomplete, so MADE only reports failures at the star-tracker-level. The result is that each of the 49 modeling elements can be used to assess system failure risks accurately, but the MADE and SysML/MagicDraw models are slightly out of synchronization (effects of this will be discussed in section 3.3).

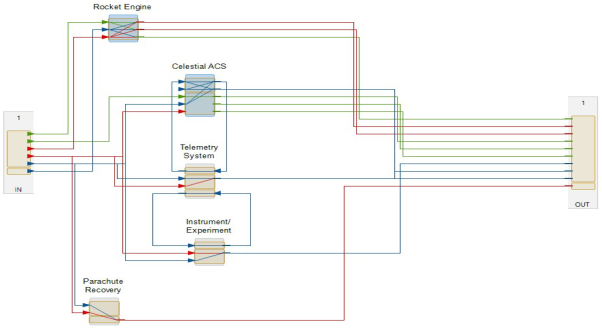


Figure 11. Sounding Rocket MADE Model

For the CapiBRIC test case, MADE ingestion resulted in 12 elements, 3 of which have underlying system decomposition modeling, and 1 has multiple layers of underlying system decomposition modeling. However, in many cases, the naming conventions exported in the BOM were excessively long since they were derived by SysML/MagicDraw from the *type's* name and hierarchy and not the name property (e.g., *"Brine Distribution Manifold"* was exported and ingested as *"Brine Distribution Manifold: CapiBRIC::ISS Design::Components::Plumbing Assembly:: brine distribution manifold*"). These types of names could have been shortened with post processing for convenience but were kept as-is to see if additional issues with using referenced architectures within a model versus libraries would be found (See 3.3 for those issues/resolutions). While cumbersome, these names did not cause any issues with assigning preliminary functions and failure concepts during the importing of each modeling element. Nor did they hinder the manual refining or adding of the assigned functions and concepts (by following the steps 4-6 and 8 in *MADE Modeling Techniques & Tips* [14] (as shown in Appendix II.B)) as needed to complete the model and represent the functions, flows, and other parameters to represent the sounding rocket system (See Figure 13) and its underlying systems' functionality and interconnectivity. However, since the legacy CapiBRIC SysML/MagicDraw IDB (Figure 12) did not clearly define the flow of material or signals through the system, developing a complete model in MADE was challenging and relied greatly on CapiBRIC documentation and the modeler's expert judgement even with the previously determined system-architecture hierarchy. This is not optimal and can be avoided with better documentation/modeling and multi-discipline modeling team communication. Further, while the SysML/MagicDraw model did not decompose the blowers of the sweep gas assembly or the two CapiBRIC humidity sensors, the NASA library does decompose these elements. So, when ingested, these decompositions became present in the MADE CapiBRIC model. The result is that each of the 27 modeling elements can be used to assess system failure risks accurately, but the MADE and SysML/Magic Draw models are slightly out of synchronization (effects of this will be discussed in section 3.3).

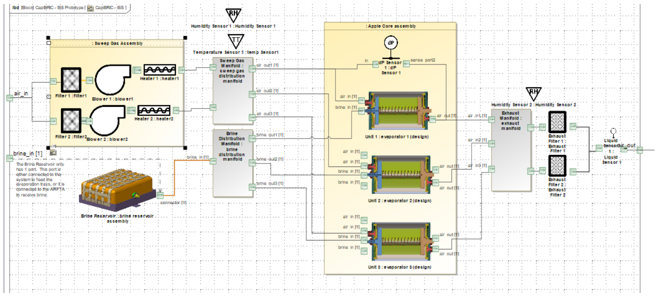


Figure 12: CapiBRIC (ISS) SysML/MagicDraw IBD

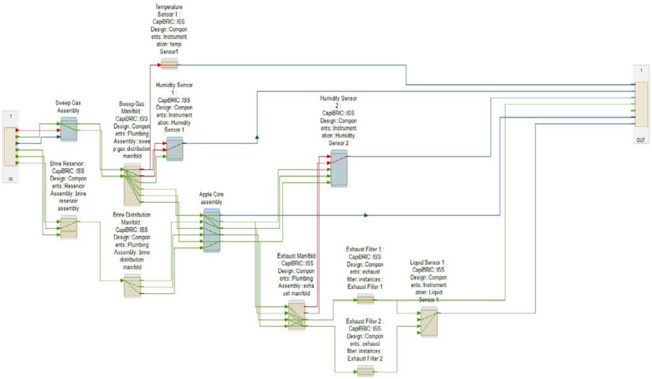


Figure 13. CapiBRIC (ISS) MADE Model

3.2.2 Modeling Verification and Validation

Modeling verification was completed by comparing the resultant MADE models (e.g., structure, connections, failure diagrams, decompositions, flows, functions) created in this study to those previously manually developed during MBSMA phases [11, 12, and 13] to ensure modeling accuracy during the completion process (See 3.2.1). Whereas modeling validation was performed after a model was completed by assessing the completeness of inductive FMECA and deductive Fault Tree outputs from this study's MADE models and comparing them to those same artifacts from MBSMAI phase 1 to ensure each model's veracity.

For each MADE model in this study, initially the generated FMECAs and Fault Tree might have had unreported systems or failure modes and would have been assessed as incomplete/unfinalized until all flow and function inconsistencies were corrected (MADE modeling corrections are unrelated to federation). Once the outputs were assessed as complete, the final validation for accuracy was completed by comparing them to the same artifacts from MBSMAI phase 1 [11] and only found disconnects from previous study outputs for those elements that were further modeled/decomposed as noted earlier. As a result, the study generated verified/

validated MADE-SMA models from SysML/MagicDraw models and FMECA and Fault Tree data for sharing with stakeholders and system models (e.g., SysML/MagicDraw models).

Note: The phase 1 ISS CapiBRIC MADE model had been limited to 1 main functional block diagram and 1 failure diagram, so FMECA/FTA artifacts from that work were of limited use.

3.3 Share SMA Data with SysML/MagicDraw Models

The following options and their efficiency, feasibility, and practicality were considered to arrive at an optimal solution for accomplishing SMA modeling federation/integration with varied SysML/MagicDraw modeling styles so that mission assurance risk assessments can be shared/communicated:

1) Transfer or replicate all SMA MADE modeling in SysML/MagicDraw. This would be complex task and would likely require new capabilities in SysML/ MagicDraw to process/ingest all the MADE features. If accomplished via custom JavaScript Object Notation (JSON) generation and ingestion methods for each SysML/MagicDraw modeling style (not a fully supported feature of or capability provided by either tool), the resulting modeling could not be used to produce SMA results with current plug-ins (e.g., FMECAs or Fault Trees from the Tietronics plug-in) nor could the modeling data be used at all unless new SysML/MagicDraw features were developed. In addition, this would also require continuous synchronization of all the MADE modeling constructs in two modeling environments/ models, if MADE capabilities (e.g., fault/anomaly simulation, RMA analysis, hazard identification/ assessment,) are to continue to be used for SMA support. Thus, this option was found to be sub-optimal and limiting in terms of current feasibility, practicality, and efficiency.

2) Transfer or replicate select subsets of the SMA MADE modeling and data (e.g., flows/functions, results data) in SysML/MagicDraw:

    a) Transferring or replicating flow/function modeling for each element from MADE to a SysML/MagicDraw IBD for model synchronization and support of other engineering disciplines: This is not doable now but will likely be doable with new capabilities being developed by PHM for MADE and SysML/Magic Draw that would be useful to not only for sync with MADE but may assist with the development/ refinement of designs/interfaces and operational concepts and may more fully complete the SysML/ MagicDraw model without extra efforts from another engineering team.

    b) Transferring or replicating SMA results data: This is doable now and not only emulates the current traditional processes, but it also leverages the best attributes of each tool to provide knowledge efficiently to stakeholders if optimized. Doing this type of transfer/replication by custom JSONs was again found not to be optimal for the same development and style compatibility reasons noted above. However, doing this type of transfer/replication via **Excel table exports and SysML/ MagicDraw table imports was found to be an optimal method**, since it leveraged existing capabilities of both tools, is compatible with any SysML/MagicDraw modeling style, only requires the addition of the federation profile for SMA data shown in Appendix B to be added to each model, presents the SMA data in user friendly manner (See Figure 14 – 19), and enables the linking of SMA data to SysML/MagicDraw model design-items via the allocation feature for alternate cross-referenceable access via an allocated-to item’s specification. Note: Transferring the data to Tietronics plug-in state machine fields was considered but was rejected due to complexity, modeling-style dependencies, and the inherent limitation of the plug-in to hold only inductive or deductive data, not both or other SMA data.

Thus, the optimal table data transfer and allocation method (See Appendix A steps III.A – III.D), which includes the simple post-processing of Excel reports to add two fields/ columns: *FMECA_ID* manual filled and *Allocate To* that can be set equal to *Component or Item* report field/column, was used for testing SMA modeling federation/integration with the varied SysML/MagicDraw modeling styles of the test cases in this study and is recommended for future modeling federation efforts. While the test cases shown below utilized only FMECA and Fault Tree data, it was noted that this methodology could be easily adapted to transfer hazard data from system safety since that data or other data from SMA modeling efforts (e.g., data quality, requirement compliance, maintainability, prognostics, predictions (reliability and availability)) can be output by MADE to an Excel report by design.

For the Europa propulsion test case, FMECA and Fault Tree data transfers created 40,387 and 63 new SysML/Magic Draw federation items respectively (e.g., failure modes, fault events (See Figures 14 and 15)), automatically allocated these to their associated modeling elements by SysML/Magic Draw matching the *Allocate To* data to a modeling element, and created a readable table for each data set. While several iterations were used to eliminate duplicated modeling names in the SysML/MagicDraw model, ultimately SysML/MagicDraw was able to make only 40,315 failure modes and 58 fault tree event allocations since the 4 REMs and the latch valve in the PIA for fuel did not match the SysML/MagicDraw element names. Note: SysML/MagicDraw will not make an automatic allocation unless it finds only a single exact name match. However, unallocated items are still available and fully populated in the table views for stakeholders to interpret and can be manually allocated, if desired (this was not done in this study so that further model synchronization can be tested in the future). While the data transfer process was successful for the Europa propulsion test case, it did require the default *heap memory allocation* to be increased from 4,000 to 10,000 (See Appendix A III.D note) to avoid SysML/Draw failures, made

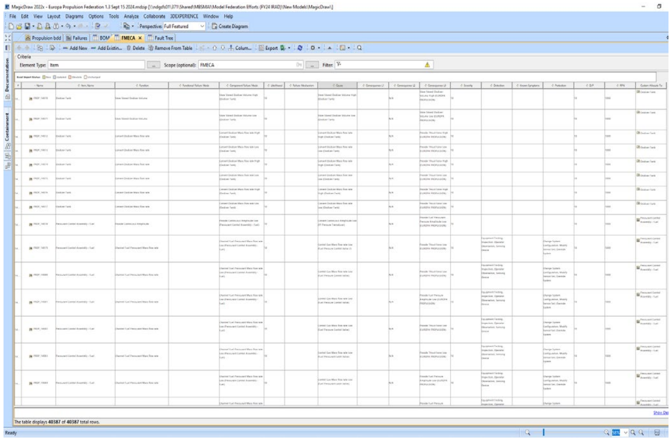


Figure 14. Snapshot of Europa Propulsion FMECA in SysML/MagicDraw

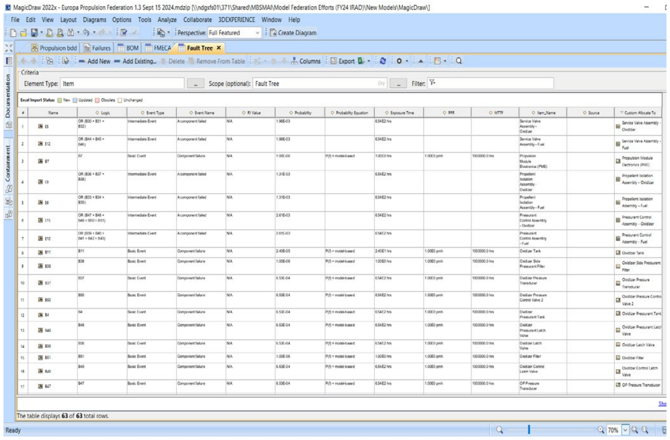


Figure 15. Snapshot of Europa Propulsion Fault Tree in SysML/MagicDraw

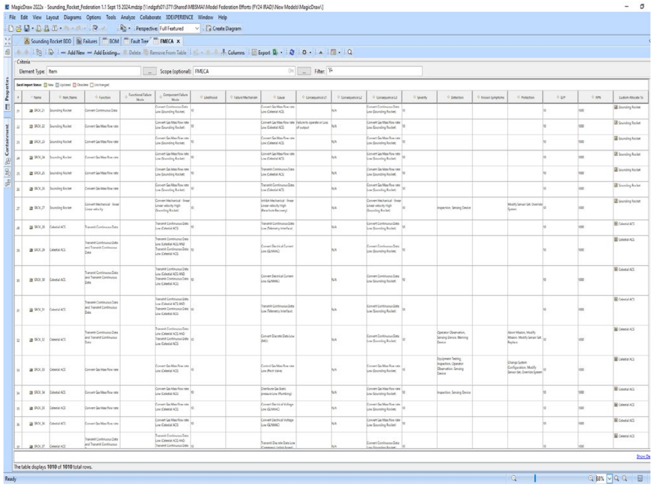


Figure 16. Snapshot of Sounding Rocket FMECA in SysML/MagicDraw

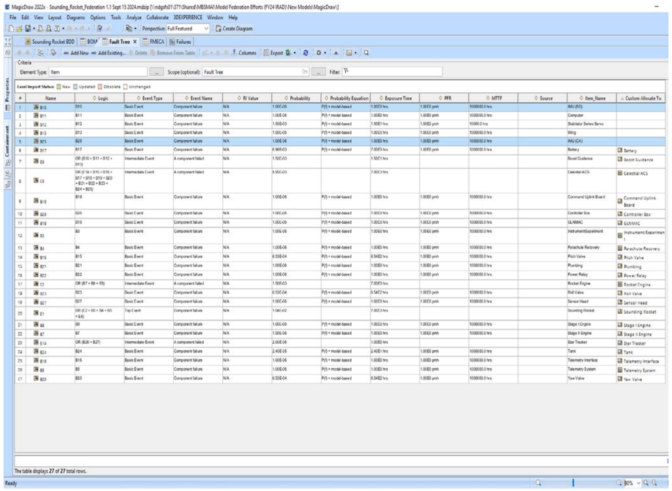


Figure 17. Sounding Rocket Fault Tree in SysML/MagicDraw

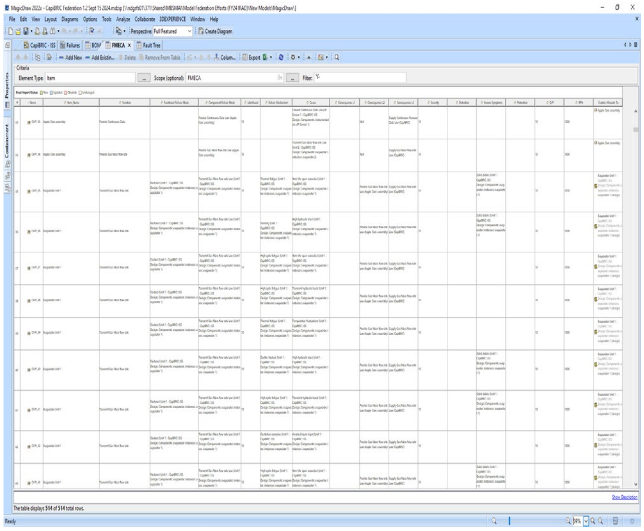


Figure 18. Snapshot of CapiBRIC FMECA in SysML/MagicDraw

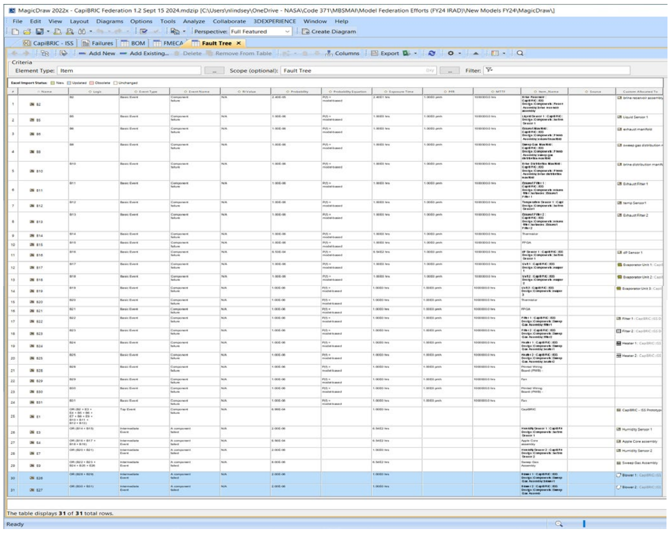


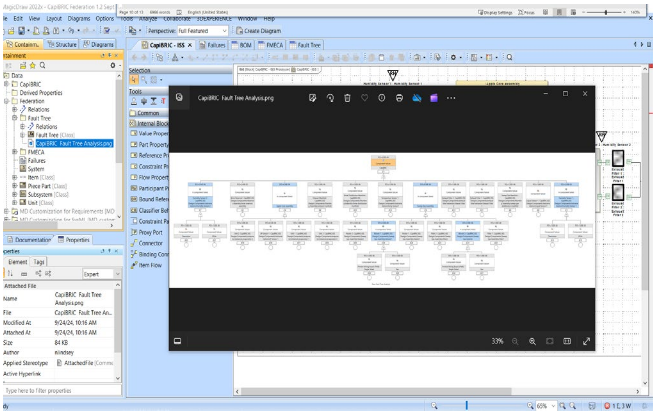
Figure 19. CapiBRIC Fault Tree in SysML/MagicDraw

the model quite large, and took multiple hours to complete since so many items and linkages were being created. Consequently, additional research/testing is recommended to identify ways to optimize the transfer process for future use and larger data sets (e.g., mission FMECAs).

For the Sounding Rocket test case, FMECA and Fault Tree data transfers created 1010 and 27 new SysML/Magic Draw federation items respectively, automatically allocated these to their associated modeling elements (as described above (See Figures 16 and 17)) and created a readable table for each data set. In this case, iterations to eliminate duplicated modeling names in the model (found mostly in packages and behaviors in this test case) and SysML/MagicDraw were only able to make 898 failure mode and 22 fault tree event allocations. This was due to the two engines and a boost guidance system with its 4 underlying systems (e.g., IMU, computer, stabilizer, and wing) enhancements (as described earlier in 3.2.1) that were made in the MADE model that did not match the SysML/MagicDraw elements names. The unallocated items were again left in that state to allow for future testing as mentioned above, but these items and their associated parameters are still fully available in the table views of the FMECA and Fault Tree. Iterations on data transfers can be avoided if a SysML/Magic Draw model uses unique names for every modeling element, even if it's a *packages* or *behavior*, therefore appending a differentiator to modeling elements with the same name is highly recommended if a more unique name is not found.

For the CapiBRIC test case, FMECA and Fault Tree data ingestion created 514 and 31 new SysML/MagicDraw federation items, respectively (See Figures 18 and 19). In this case, iterations to eliminate duplicated modeling names in the model were much more complex and numerous since, as mentioned earlier, a referenced architecture model included in the model and used to define the ISS implementation model of the test case used the exact same names, and the lengthy names (*Type* name as mentioned above) exported to MADE were not the *names* the allocation function uses for its matching. Thus, the Excel exports had to be post-processed again to manually assign a matching item name for allocation to each failure mode and fault event. Nevertheless, SysML/MagicDraw still sometimes matched the ISS-implementation-matching name back to the referenced architecture instead (this could not be overcome) due to the referenced style of the model. Ultimately, SysML/MagicDraw was only able to make 448 failure mode and 23 fault tree event allocations (to the ISS implementation and the referenced architecture) since the blowers of the sweep gas assembly were decomposed and the two missing CapiBRIC humidity sensors were added to the MADE model as described above in section 3.2.1. The significant iterations (and additional post-processing of the Excel report) needed for successful allocations and the confusing allocations can be avoided if the SysML/MagicDraw model does not use referenced architectures within a model (this is highly recommended) but instead uses libraries or other means to create consistent elements between multiple implementations of the same design without using cross-referencing and duplicate names.

It is important to note that while the FMECA tables mentioned above, are nearly an exact duplicate of the traditional tables included in a FMECA report (and provide a traditional view to stakeholders), the Fault Tree tables are not the traditional graphical view but provide the same information. However, this can be overcome now by capturing the MADE-provided pictorial output and adding it to a model under the Federation area using Appendix A step III.D.3 (see Figure 20). Note: Graphics will open in their native program not SysML/ MagicDraw. It is plausible that in the future a plug-in/view could be developed, if desired, to read the Fault Tree table and create a more integrated graphic in the unlikely event that is needed.



Name: CapiBRIC Fault Tree Analysis.png
File: CapiBRIC Fault Tree An...
Modified At: 9/24/24, 10:16 AM
Attached At: 9/24/24, 10:16 AM
Size: 84 KB
Author: nlindsey
Applied Stereotype: AttachedFile [Comme
Active Hyperlink:

Figure 20. Fault Tree Graphic Inclusion/Display Example

4 *SUMMARY & CONCLUSIONS*

4.1 Findings/Discoveries

As stated earlier, this study was initiated to determine and test the feasibility of implementing interoperability solutions between MB-SMA-modles (MADE) with other project disciplines via direct collaboration with DE/MBSE-models (SysML/MagicDraw). It intended to and has tested bidirectional synchronization/data transfers at the subsystem-level and found:

- ✓ It is possible to transfer/incorporate DE/MBSE SysML/MagicDraw system elements (e.g., BOM elements) in MADE to create a MBSMA model and add SMA characteristics (e.g., failure concepts and functions); and is relatively simple to achieve by using of SysML/MagicDraw tables for exporting the BOM data to Excel and using MADE import functions with NASA and MADE Pallete libraries to establish preliminary functions and failure concepts that can be modified or customized to completely model any system.
- ✓ It is possible to transfer/incorporate SMA/MBSMA model findings (e.g., failure modes, fault events) into a DE/MBSE system model and emulate/attach artifacts (e.g., FMECA, Fault Tree) and is relatively simply to achieve (and can be expanded to other SMA artifacts/findings like hazard analysis) by using

MADE's extensive Excel and pictorial output options and SysML/MagicDraw tables and the *Allocate To* attribute.

This study also revealed that while both SysML/MagicDraw and MADE can fully support model data federation via the procedures and processes recommended by this study (See Appendix A and Section 4.2), other tools (model-based or not), delivered products (e.g., vendor or partner submittals), or traditional methods may also be able to provide SMA (or even BOM) findings (in Excel) for data and model federation as well with only potentially minor adjustments to the procedures and modeling constructs used in this study. As this may introduce additional synchronization issues, this should be done with care and the knowledge that re-processing to match modeled configurations and item names is likely to be required.

4.2 Guidance & Recommendations

Based on the results, findings, and the data transfer/modeling experiences of this study, the following is recommended:

1) Modelling teams modify and expand processes and controls recommended in previous studies [11,12, 13] as follows: 1) Establish a multi-discipline modeling team (Systems Engineering (SE) and Safety and Mission Assurance (SMA) at a minimum); 2) Establish modeling responsibilities (e.g., SE's model requirements and system configuration hierarchy in BDDs, REs model failure behaviors/characteristics/controls and codify, with input from the design and systems teams, configuration element-functionality and interface/flow details get included in Functional Block/Wire Diagrams, Safety Engineers model hazards); 3) Complete SMA-modeling, based on importing/receiving modelling elements and data from the system model by (3A) creating a BOM in the system model and exporting that to Excel, (3B) importing the exported BOM into the SMA-model, and (3C) performing SMA modeling; 4) Produce SMA artifacts and share resulting data between modelling elements by (4A) exporting SMA artifacts and artifact-data items in Excel and (4B) ingesting/emulating those artifacts in the system model in tables and attachments; 5) Validate and refine modelling (and designs) until a final and acceptable result is achieved; and 6) Share modeling with future missions.

2) Modelers ensure system SysML/MagicDraw models include a hierarchical system decomposition in a BDD with consistent ownership relationships and application of federation stereotypes (e.g., system, subsystem, unit, piece part) so a BOM table for federation and other purpose can be created. This BDD can also be used for requirement allocation and work breakdown structuring.

3) During the establishment of roles and responsibilities, integrated modeling teams agree that SMA will model/codify, with input from the design and systems teams, the additional functional details to conduct SMA analysis/assessments. This is more than traditional block or interface diagrams would provide (e.g., Propulsion Latch Valves would need power/command inputs and telemetry outputs as well as material flows defined to fully characterize their functionality) and would elevate understanding of other mission-development teams as well. These characterizations can be shared back to the system SysML/MagicDraw model as a graphic (as is the current/traditional project communications method) and soon will also be shareable as an Interface Block Diagram (IBD) with the development of MADE functional diagram to SysML/MagicDraw IBD transfer capability.

4) Modelers ensure or standardize that clear and differentiable naming of modeling elements is used in all models (no two modeling elements, packages, behaviors, or state machines should share the same name) and libraries are used for commonly used architectures or elements to avoid having cross-referencing and allocation issues.

5) Integrated modeling teams use the data federation procedure (Appendix A) developed during this study, the federation supporting profiles/stereotypes shown Appendix B, and MADE to complete SMA analysis/assessment integration with system SysML/MagicDraw models. The use of MADE is recommended since it can support SMA, has been evaluated to have the best model-based SMA Return on Investment (ROI) based earlier study phases [12, 13], and can produce Excel reports/exports of hazard, data quality, requirement compliance, prognostics, reliability (FMECA and Fault Trees), prediction (availability and reliability (RBDs)), and maintainability analysis/assessment. While federation avoids the significant SysML/MagicDraw workload and resultant inconsistencies in manually creating 20-30 state machines and behaviors for capturing each subsystem's failure signatures that are only either inductive or deductive signatures, it also eliminates the need to create two SysML/MagicDraw models for producing an accurate FMECA and Fault Tree [15] since FMECA and Fault tree modeling would not (and should not) be done in SysML/MagicDraw.

6) Integrated modeling teams use the portion of the data federation procedure for transferring/incorporating SMA/MBSMA model findings (e.g., failure modes, fault events) (Appendix A) and the federation supporting profiles/stereotypes (Appendix B), developed during this study, for any SMA analyses or assessments (model-based or not) that can be output/delivered in Excel. But great care and potential re-processing to match modeled configurations and item names should be used.

7) The MBSMAI team continues the currently planned additional federation and model-synchronization testing, optimizations, and developments to:
    - Determine a method for sharing MADE functions and flows and configuration updates/decompositions with SysML/ MagicDraw to update MBSE model's containment tree (structure and content) and BDD; and create/replace IBD.
    - Determine a method for sharing SysML/MagicDraw configuration updates/decompositions with MADE without updating the entire model.
    - Identify and test additional large-table ingestion methods in SysML/MagicDraw to avoid excessive wait times.
    - Identify and resolve any federation issues at the mission-level, including but not limited to the potential for system models to grow quite large and slow SysML/MagicDraw performance.
    - Test the federation of data from other model-based SMA-discipline outputs that have been tested (and non-model SMA data imports) to verify assumed capabilities.
    - Expand MBSMA to include Software Assurance and Quality Engineering.

4.3 Conclusions

Thus, this study concludes that Model federation is easy, valid, and plausible between Mission Assurance and Engineering models if adequate modeling processes, procedures, and compatible modeling styles/structures are established and implemented. Further, federation will remain straightforward and efficient if the recommendations provided herein are adopted by all modeling teams.

5 PATH FORWARD

The provided results, findings, guidance, and recommendations were formulated based on subsystem model testing only but can be used now by NASA/GSFC to advance digital transformation efforts in general and to assist in the establishment of SMA-to-SE and SE-to-SMA modeling data transfers, collaborations, and transition points. However, to ensure that the solutions and procedures of this study are scalable to mission-level modeling/assurance efforts and other SMA disciplines, GSFC plans to execute a FY25 IRAD to perform mission-level federation testing, expand/test the feasibility of SMA modeling including Software Assurance (SWA) efforts (Quality Engineering testing is planned for FY26), continue to maintain the SMA-reference model to jumpstart new modeling, and continue to share modeling tips/guidance broadly.

REFERENCES

[1] "Goddard Procedural Requirements for Risk Management; GPR 7120.4D," Code 300/Safety and Mission Assurance Directorate, Goddard Space Flight Center, Greenbelt, MD, USA, 2012.

[2] "Systems Modeling Language," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Systems_Modeling_Language. [Accessed 15 Apr 2019].

[3] "MADE Software Official Website," PHM Technology, [Online]. Available: https://www.phmtechnology.com/. [Accessed 21 Jan 2019].

[4] "NASA Sounding Rockets User Handbook," Sounding Rockets Program Office, Sub-orbital and Special Orbital Projects Directorate, Wallops Island, VA, USA, 2015.

[5] M. J. Sargusingh, S. Pensinger and M. R. Callahan, "CapiBRIC – Capillary-Based Brine Residual In-Containment for Secondary Water Recovery," EISD Technology Showcase, 2015.

[6] M. R. Callahan and M. J. Sargusingh, "Design Status of the Capillary Brine Residual," in 46th International Conference on Environmental Systems, Vienna, Austria, 2016.

[7] S. Friedenthal, R. Griego and M. Sampson, "INCOSE Model Based Systems Engineering (MBSE) Initiative," in INCOSE 2007 Symposium, San Diego, CA, USA, 2007.

[8] "EUROPA Propulsion Critical Design Review Package (EUROPA-PROP-REVW-0006)," GSFC EUROPA Project, Greenbelt, MD, USA, 2018.

[9] NASA/OSMA, "Safety and Mission Assurance," accessed May2024, https://www.nasa.gov/safety-and-mission-assurance/

[10] G.K. Gafka, "Safety and Mission Assurance Overview, " University of Colorado/JSC, Houston, TX, 2011, https://ntrs.nasa.gov/api/citations/20110015823.

[11] N. J. Lindsey, M. Alimardani and L. D. Gallo, "Reliability Analysis of Complex NASA Systems with Model-Based Engineering," GSFC Risk & Reliability Branch, Greenbelt, MD, USA, 2020, https://ntrs.nasa.gov/api/citations/20200000582/

[12] N. J. Lindsey, M. Alimardani and L. D. Gallo, "GSFC/NASA Model-Based SMA Initiative Phase 1 Report: Reliability," GSFC Risk & Reliability Branch, Greenbelt, MD, USA, 2019, https://ntrs.nasa.gov/citations/20205009990.

[13] N. J. Lindsey, M. Cooter, A. Hodges, and N. Sindjui, "System Safety Analysis of Complex NASA Systems with Model-Based Engineering," GSFC SMA, 2022, https://ntrs.nasa.gov/citations/20220011346.

[14] N. J. Lindsey, M. Alimardani and L. D. Gallo, "MBSMAI Phase 1 Recommended MADE Modeling Techniques & Tips," GSFC Risk & Reliability Branch, Greenbelt, MD, USA, 2021, https://ntrs.nasa.gov/api/citations/20210000720/downloads/MADE%20SMA%20Enabling%20Techniques%20and%20modeling%20Tips%20from%20MBSMAI%20Phase%201%20.pdf

[15] N. J. Lindsey, M. Alimardani and L. D. Gallo, "MBSMAI Phase 1 Recommended SysML/Magic Draw Modeling Techniques & Tips," GSFC Risk & Reliability Branch, Greenbelt, MD, USA, 2021,https://ntrs.nasa.gov/api/citations/20210000720/downloads/SysMLMagicDraw%20SMA%20Enabling%20Techniques%20and%20modeling%20Tips%20from%20MBSMAI%20Phase%201%20.pdf

ACKNOWLEDGEMENT

We thank the GSFC IRAD team for sponsoring this effort and Rahul Gottumukkala (PHM), Aron Comis (GSFC) and Rob Morgenstern (GSFC), for their support and modeling tool insights, assistance, and guidance which were instrumental to this study's success.

BIOGRAPHIES

Nancy J Lindsey, Goddard Space Flight Center (Code 371)
8800 Greenbelt Road, Greenbelt, MD 20771
e-mail: nancy.j.lindsey@nasa.gov

Nancy J. Lindsey has spent 40 years in aviation and aerospace engineering performing a variety of reliability, assurance, and systems engineering tasks across the entire gamut of space vehicle life cycles and program types including Defense and Commercial Missions, Space-based Astronomical Observatories, and Earth Science Monitoring Systems. She is a recognized innovator and is currently a SMA Staff Engineer specializing in RMA/EOMP. She has a Bachelor of Science degree in Computer Science & Aeronautical Engineering and a Master of Science degree in Space Studies. Nancy's independent research efforts can be viewed via website: www.rcktmom.com.

Nobel Sundjui, Goddard Space Flight Center (Code 371)
8800 Greenbelt Road, Greenbelt, MD 20771
e-mail: lionel-nobel.w.sindjui@nasa.gov

Lionel-nobel Sindjui received a BS in Mechanical Engineering in 2015, and a Master of Engineering in Mechanical Engineering in 2018, both from the University of Maryland College Park. He is currently working as Reliability Engineer at NASA Goddard Space Flight Center (GSFC). During his time at NASA GSFC, he has worked continuously to improve the printed circuit board assurance process and test the capacities of Model Base Engineering tools for NASA missions. This includes providing finite element analysis techniques to simulate Printed Circuit Board (PCB) environmental conditions and modeling analysis to test the design architecture of NASA projects.

I *Generate and Export Bill of Materials (BOMs)*

A. Create a hierarchical representation of the system in SysML/MagicDraw *Containment Tree* and *BDD*.

1) Select *Model* in containment and right click to *Create Element* and create new *packages* and *blocks* that represent system.
2) Select *Diagrams* and right click to *Create New Diagram* and create a BDD that show the systems hierarchy using *owned* attribute in *Specification* window.

B. Apply Federation stereotype classes (System, Subsystem, Unit, or Piece Part) as appropriate to each item in the *Containment Tree* and *BDD*.

1) Apply Federation package/profile to model or create it.

   1.1 Copy Federation profile from a source model and paste it in model under *Model* in containment tree;

   Or

   1.2 Select *Model* in containment and right click to *Create Element* and create a new package called 'Federation' then select *Federation* in the containment tree and right click to *Create Element* and create the new stereotype classes, circled on right, and profile shown in the red boxes on the right.

2) Open *Specification* window for each element in the BDD and assign the appropriate stereotype.

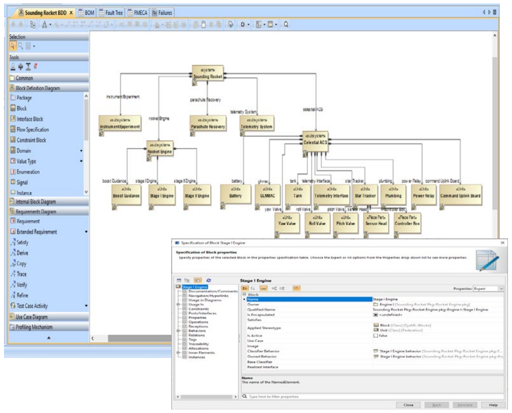
C. Create or use reference BOM Diagram.

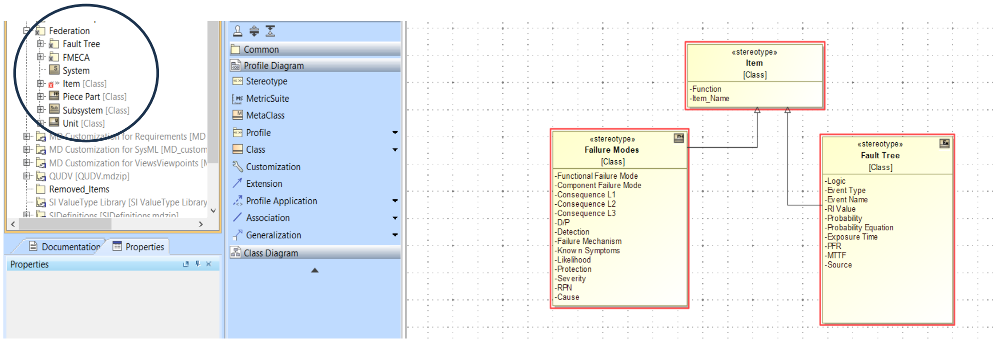
1) Copy a BOM diagram from a source model and paste it in model under *Model* in the containment tree;

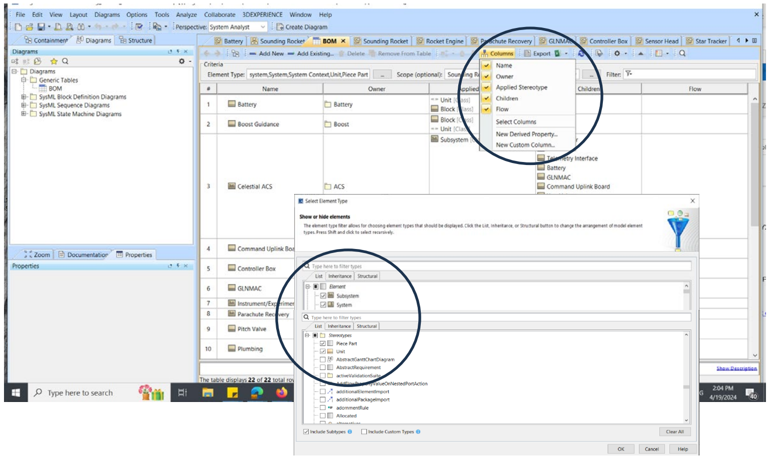
Or

1) Select *Model* in containment and right click to *Create/Add New Generic Table Diagram* called 'BOM'; Then select *Columns* by checking *Name, Owner, Applied Stereotype, Children,* and *Flow,* circled on right; And set *Element Type* to Federation Stereotypes using three dot button and checking *System, Subsystem, Unit,* and *Piece Part,* circled on the right.
2) Set *Scope* on BOM diagram to the top or desired system modeling element in containment and table will auto-populate.

D. Export BOM Table to Excel using *export table menu option* and save file.







II *Ingest Modeling Elements and Finalize SMA Modeling*

A. Ingest the exported BOM modeling elements into MADE.

1) Post-process exported Excel to get each parent-child relationship represented on a separate line.

1.1 Make a copy of the *Report* tab and adjust row height in new tab to see all children/data listed. (Optional: Hide Owner/Stereotype columns)

1.2 Create a row for each parent-child relationship by

- Selecting the *Name* and *Children* columns and use *Data* menu and the *From Table/Range* option in Excel with ignore header checked; select OK to open *Table Query* feature.
- Selecting any cell with Children and select *Split Column - By Delimiter* drop-down; Clear options (select custom); and *Select Advanced Options* and choose *Spilt into Rows* and by *Special Character* of line feed.
- Clicking *Close and Load,* and a Table will now be added to Excel file.

1.3 Name new tab (e.g., '*Table 1*'), make Column E = to Column D for each item, and save file.

2) Ingest Excel BOM in MADE

2.1 Open MADE and create a new project.

2.2 Select *Preferences- Application Preferences – Modeling – Import – Component* in MADE menu and create or modify *model import configuration* as follows, then *Apply/Close* (study used 'NASA' as name):

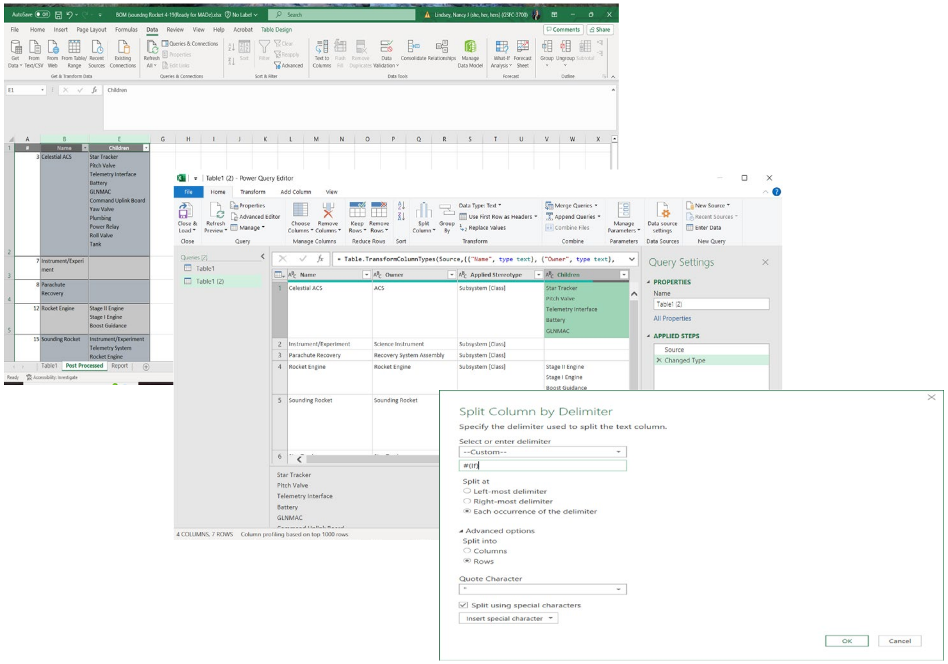
- Set *Name* = "NASA", or name used above
- Set *Worksheet* = tab name (e.g., Table 1)
- Set *Header Rows* = 1
- Set *Parent Item Number* = Column A
- Set *Item Number* = Column D
- Set *Item Name* = Column E

2.3 Use the MADE *File* menu to select *Import* to open *Import Wizard* and specify Excel spreadsheet file name and select *Model Items*. The *Model Import Configuration* can be verified or updated using *New* and editing the parameters. Click *Next.*

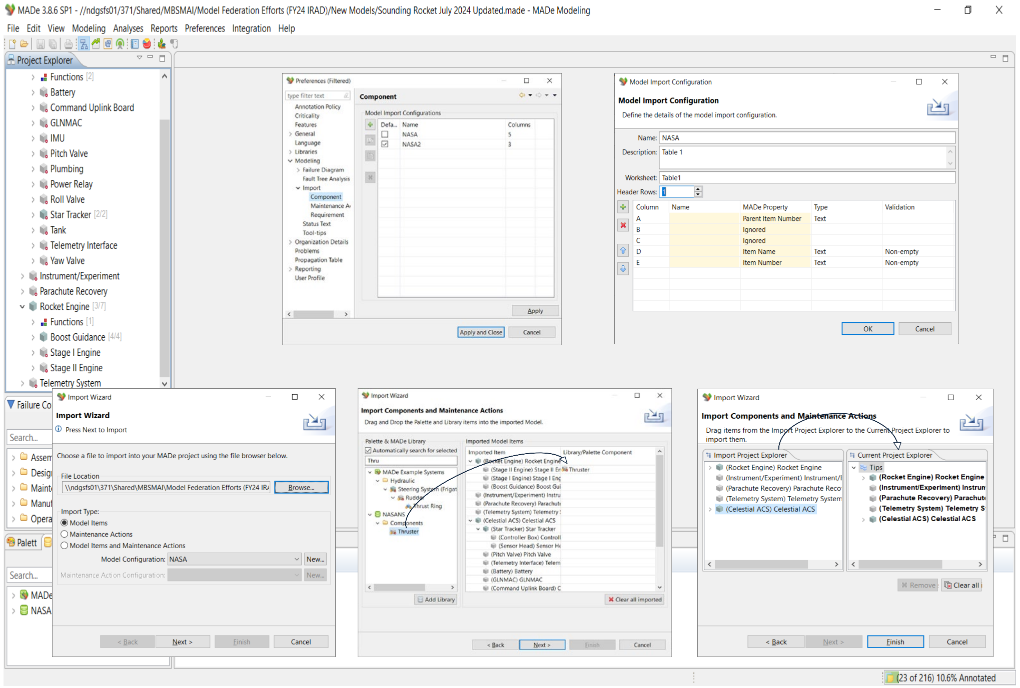
2.4 Ignore and *Close Imported File Problems* window. Click *Next* on *Imported Items* window (this is just informational), then match all possible items with *pallete* or *library* items (this is a one-at-a-time and one-time step and can't be done after import but can be revised before importing).

2.5 Click *Next* and select all items (collapsed to subsystem for simplicity) on the left (*Import Project Explorer*) and drop them in MADe top-level file folder on right side (*Current Project Explorer*) of *Import Wizard*.

2.6 Click *Finish,* and the importing will be completed, and the project will be populated.



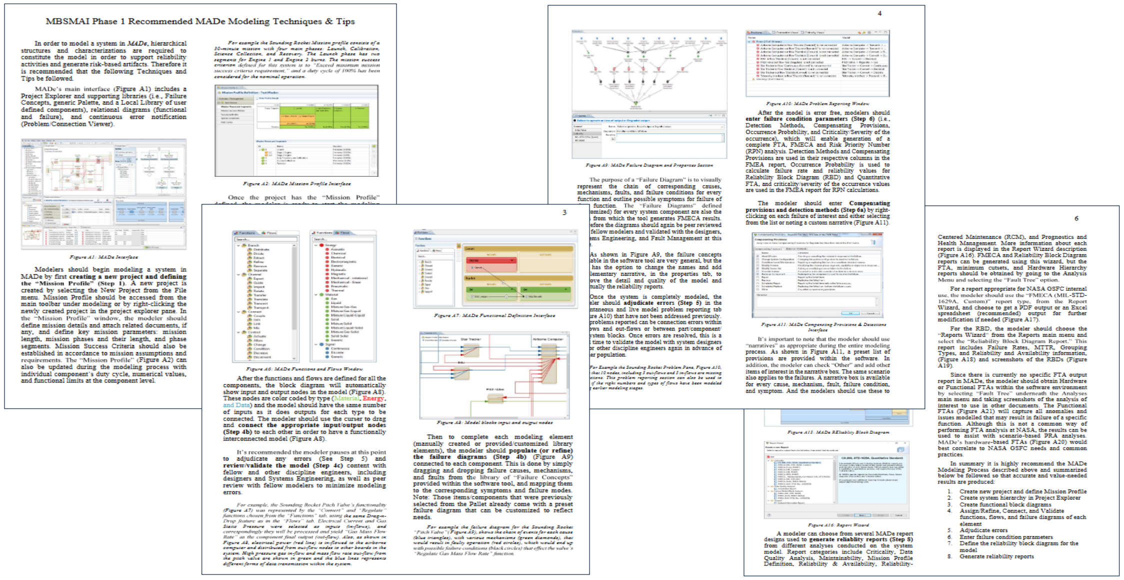
BOM Excel Post-Processing



MADE Ingestion Screens

B. Complete modeling for MA analysis/support as needed and iterate as necessary.

1) Add *Mission Profile* (optional) following step 1 of NTRS – 20210000720 MADE tips.
2) Refine, connect, and validate *Functions/Flows and Failure Diagrams* (or other characteristics) to capture the behavior of each element of the system, following step 4 of NTRS – 20210000720, MADE tips. These may need to be assigned if not matched during importing.
3) Enter system or design specific condition parameters (i.e., Detection Methods, Compensating Provisions, Occurrence Probability, and Criticality/Severity of the occurrence) and supporting narratives, following step 6 of NTRS – 20210000720, MADE tips. Note: A *Narrative Box* is available for these parameters (and every cause, mechanism, fault, failure condition/mode, and symptom) and should be used to contextualize outputs for each application.
4) Ensure modeling accuracy of model with built-in error-checking (see step 5 of NTRS – 20210000720, MADE tips) and by generating/reviewing SMA artifacts/outputs (see Step 8 of NTRS – 20210000720, MADE tips) for appropriateness and completeness.


NTRS – 20210000720, MBSMAI Phase 1 Recommended MADE Modeling Techniques & Tips Excerpts Only

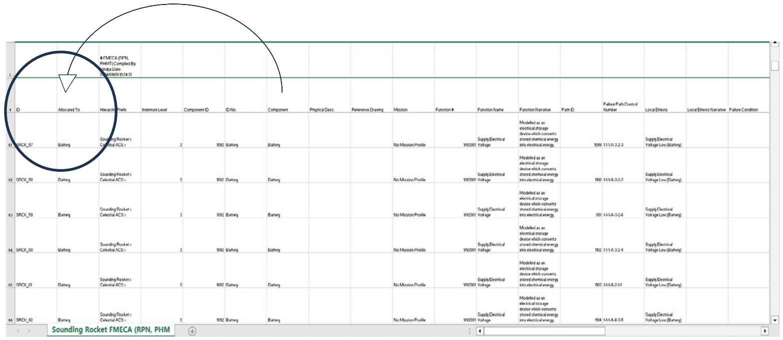
III *Share SMA Data with SysML/MagicDraw Models*

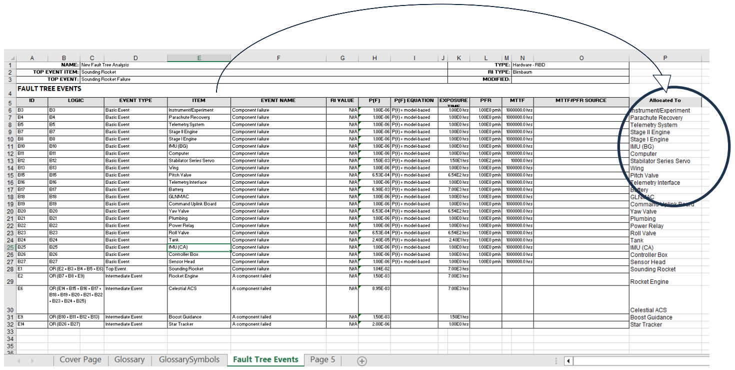
A. Generate, export, and save SMA results/artifacts in Excel and pictures (or other desired attachments).

B. Post-process exported Excel(s) (generated or received from source other than MADE) to ensure that there are sufficient identifiers and configuration item name repeats to enable SysML/MagicDraw ingestion as SMA results as tables (e.g., FMECA, Fault Tree).

1) For the FMECA Excel (from MADE), add two columns to the left of the output data; label the first column *ID* and populate it with a unique identifier for each failure mode; label the second column *Allocate To* and equate it to or copy it from the *Component* column.
2) For Fault Tree report in Excel (from MADE), add one column to the right of the output data; label that column *Allocate To* and equate it to or copy it from the *Item* column. The *ID* in the report is unique by default, but if multiple Fault Trees are planned to be uploaded to SysML/MagicDraw, then another user-defined *ID* column (e.g., system_*MADE-ID*) may need to be added.

Note: this post-processing may change if Excel is not from MADE, or another SMA data set is to be uploaded.


FMECA Excel Report Post-Processing

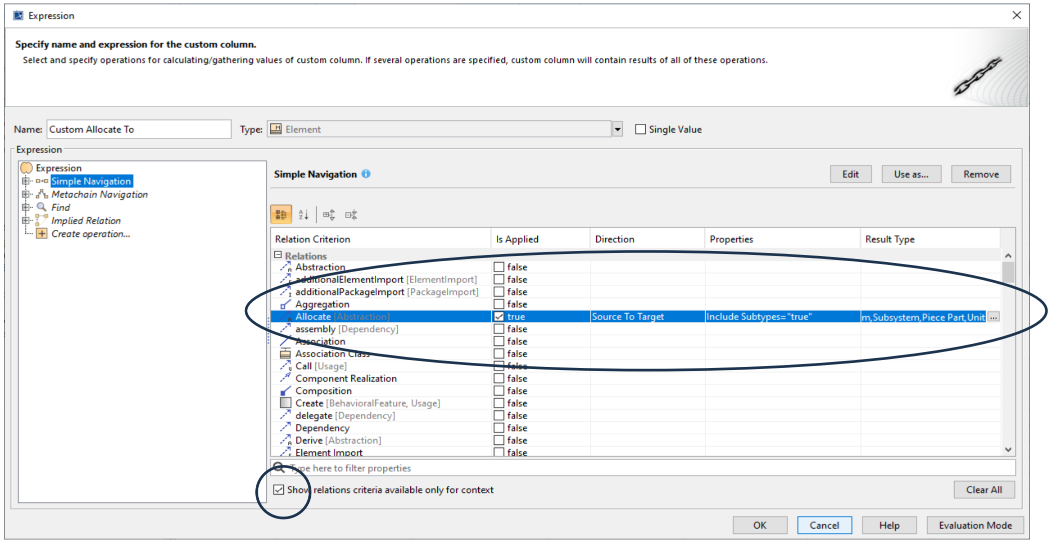

Fault Tree Excel Report Post-Processing

C. Add a generic table for each SMA results set under the Federation packages of the same name or under related system (tables can also be moved after they are created); or copy a table from another model and skip to step D.

1) For a FMECA, select the *FMECA* package under *Federation* in the containment tree and right click to *Create/Add New Generic Table Diagram* called 'FMECA'; Then Select *Columns* by checking the *Item* package stereotypes of *Item* and *Function* and all *FMECA* package stereotypes except *Item_Name* and *Function* since they are already coming from *Item_Name*, order as desired or as prescribed by stakeholders/best practice*,* see image on the lower right.

2) For a Fault Tree, select the *Fault Tree* package under *Federation* in the containment tree and right click to *Create/Add New Generic Table Diagram* called 'Fault Tree;' Then Select *Columns* by checking the *Item* package stereotypes of *Item* and *Function* and all *Fault Tree* package stereotypes except *Item_Name* and *Function* since they are already coming from *Item_Name*, order as desired or as prescribed by stakeholders/best practice*,* see image on the lower right.

3) And in all cases, also add a custom column called *Custom Allocate To* that's configured (see image and circle on the right) and has *Result Types* limited to the Federation Stereotypes *System, Subsystem, Unit,* and *Piece Part,* using that field's three dot button and selecting only those stereotypes, see oval on the right.


Expression

Specify name and expression for the custom column.

Select and specify operations for calculating/gathering values of custom column. If several operations are specified, custom column will contain results of all of these operations.

Name: Custom Allocate To

Type: Element

☐ Single Value

Relation Criterion	Is Applied	Direction	Properties	Result Type
Relations				
Abstraction	☐ false			
additionalElementImport [ElementImport]	☐ false			
additionalPackageImport [PackageImport]	☐ false			
Aggregation	☐ false			
Allocate [Abstraction]	☑ true	Source To Target	Include Subtypes="true"	m,Subsystem,Piece Part,Unit
assembly [Dependency]	☐ false			
Association	☐ false			
Association Class	☐ false			
Call [Usage]	☐ false			
Component Realization	☐ false			
Composition	☐ false			
Create [BehavioralFeature, Usage]	☐ false			
delegate [Dependency]	☐ false			
Dependency	☐ false			
Derive [Abstraction]	☐ false			
Element Import	☐ false			

Type here to filter properties

☑ Show relations criteria available only for context

SysML/MagicDraw *Custom Allocate To* Specification

D. Ingest SMA-Excel-artifacts and attachments into SysML/MagicDraw.

1) On the SysML/MagicDraw SMA-Table to be populated under the *Excel drop-down,* select *Sync Options* and
    - Select *From File System* and enter/browse to post-processed Excel file to be ingested.
    - Select *Delete elements from model* under *Sync Options* to avoid duplications.
    - Under *Mapping* enter correct sheet name and first cell for data headings, leave *CVS delimiter* and *Identification Property* unchanged, check *first row contains headings*, and match Columns from the SMA-Table to the Excel file to be read, as shown on the right. Click OK.

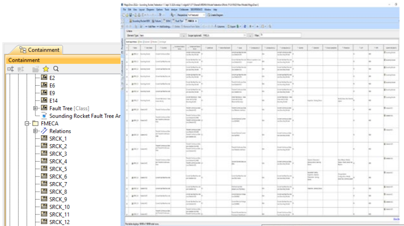
2) On the SysML/MagicDraw SMA-Table to be populated under the *Excel drop-down*, select *Read From File* and the table will populate, new items will appear under corresponding packages, and allocations will be in specification of allocated-to item and the SMA items.

3) Add SMA artifact-related attachments (e.g., Fault Tree image from MADE, see oval and image on the right) by dragging and dropping them into the corresponding SMA package (e.g., FMECA, Fault Tree).

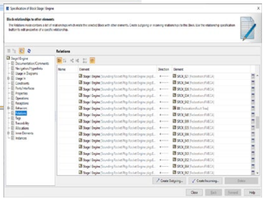
Note: To accomplish large ingestions or run big models, the *Heap Memory Allocation* for SysML/MagicDraw should be enlarged from the default 4,000 to 10,000 by following the instruction under the first option, "To change the amount of allocated memory in the Environment Options dialog" here: https://docs.nomagic.com/display/MD2022xR1/Memory+allocation.

NASA GSFC Column Order	SysML/MagicDraw Column Names	Post-Processed SMA-Excel-Artifact Column Names
1	Name	ID
2	Item_Name	Component
3	Function	Function Name
4	Functional Failure Mode	Fault
5	Component Failure Mode	Local Effects
6	Likelihood	Occurrence
7	Failure Mechanism	Mechanism
8	Cause	Cause
9	Consequence L1	Failure Condition or Narrative
10	Consequence L2	Next Effects or Narrative
11	Consequence L3	End Effect or Narrative
12	Severity	Severity (AB or AK)
13	Detection	Failure Detection Method or Narrative
14	Known Symptoms	Symptoms
15	Protection	Failure Comp Provision or Narrative
16	D/P	Detectability
17	RPN	RPN
18	Custom Allocated To	Allocate To

FMECA Sync Option Settings



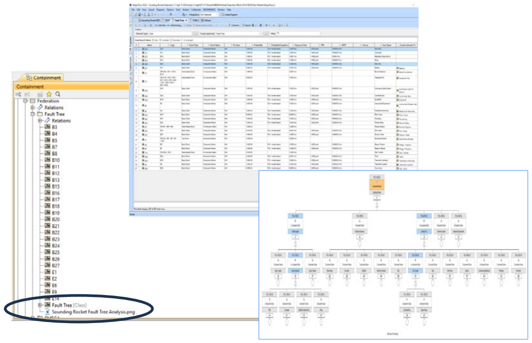
FMECA (Table and Items) in SysML/MagicDraw



FMECA and Fault Tree Allocations in SysML/MagicDraw

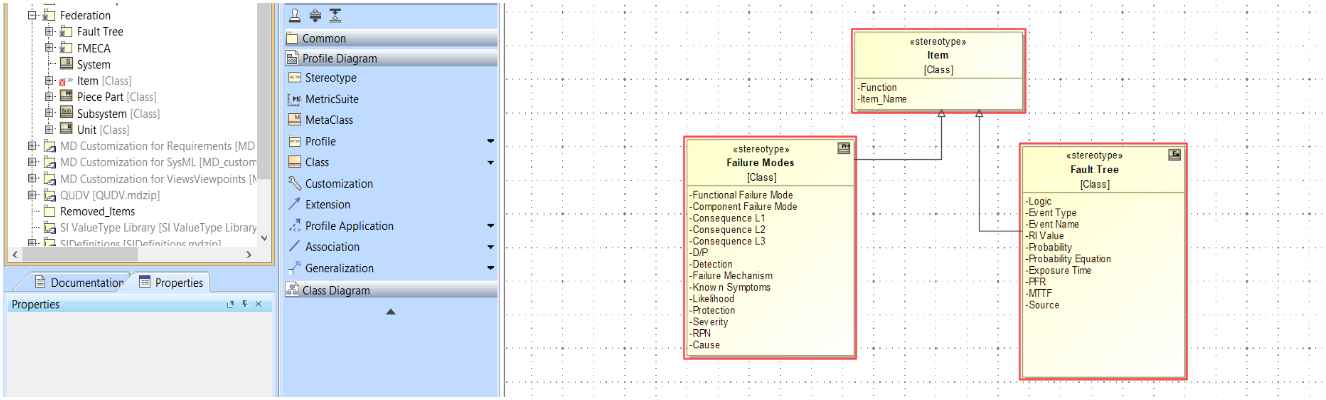
NASA GSFC Column Order	SysML/MagicDraw Column Names	Post-Processed SMA-Excel-Artifact Column Names
1	Name	ID
2	Logic	Logic (1st instance)
3	Event Type	Event Type
4	Event Name	Event Name
5	RI Value	RI Value
6	Probability	P(F)
7	Probability Equation	P(F) Equation
8	Exposure Time	Exposure Time
9	PFR	PFR
10	MTTF	MTTF
11	Item_Name	Item
12	Source	MTTF/PFR Source
13	Custom Allocated To	Allocated To

Fault Tree Sync Option Settings



Fault Tree (Table, Items, and Image) in SysML/MagicDraw

Appendix B: Federation Profile



Copy the above Package from a reference model or create a Federation Package as follows:

---

1. In the Model Browser, right-click the root *Model* or *System* and create a new Package. Name it *Federation.*
2. Right-click on *Federation* and create a new *Profile Diagram*. Name it *Failures,* as used in this study and shown above, or *SMA* for more universal applications (this will be used by GSFC-MBSMA in future applications).
3. Create *Class* stereotypes (in the diagram (e.g., *Item, Failure Modes, Fault Tree*)) and use Generalizations between them to relate one to the other.
4. Add properties using the *Create Property* plus-sign-circle (to the top right of each stereotype box) that match the data fields of each SMA artifact to be ingested (e.g., *Detection, Logic*) with a "-" in front of the field name so it is recognized as a part of the stereotype.
5. Use Create *Class* stereotypes for *System, Subsystem, Unit,* and *Piece Part*.
6. Save the project.

Now you may use this package/profile in other projects.