MBSE Applications for the MSR SRC Mars Ascent Vehicle

Isabeta Rountree
Sev1Tech LLC
12700 Black Forest Ln Ste 306
Woodbridge, VA 22192
isabeta.e.rountree@nasa.gov

Abstract— **The objective of the NASA Mars Sample Return (MSR) Campaign is to collect samples from the surface of Mars and return them to Earth for scientific research. The Mars Ascent Vehicle (MAV) will be integrated into a larger Mars Sample Retrieval Lander (SRL) for transit to and storage on Mars. After all Martian samples have been collected and loaded into the MAV payload assembly, MAV will deliver the samples from the Martian surface to Mars orbit. A separate spacecraft, the Earth Return Orbiter (ERO) will retrieve the samples from Mars orbit and return them to Earth.**

To address common systems engineering challenges associated with using traditional systems engineering practices on complex projects, the MAV systems engineering team has explored implementation of Model-Based Systems Engineering (MBSE) tools and languages. This paper describes the current state of implementation and development of the MAV MBSE model with the Systems Modeling Language (SysML) within the scope of the MAV Systems Requirement Cycle (SRC) systems engineering workflow. The MAV MBSE model has been developed within Magic Draw – a SysML editor commonly used to implement MBSE. The MAV MBSE model has been used to develop mission phase functional flow diagrams for the Concept of Operations, decompose mission to vehicle subsystem functions, develop a functional decomposition, derive functional requirements, trace requirements up to customer-imposed requirements, trace requirements within MAV requirement space, identify requirements trace gaps, define and map the physical design space architecture, allocate requirements to subsystems, develop validation items, define assembly, integration, and test (AI&T) operations, and trace these items across driving goals to develop an integrated digital thread of systems engineering information used to drive design specifications, decision making, and ultimately design verification and validation. Findings and results associated with implementing MBSE in these ways, alongside traditional methods will be discussed.

TABLE OF CONTENTS

1. INTRODUCTION

As no launch vehicle has previously taken off from another planet, design of the Mars Ascent Vehicle (MAV) has presented many challenges. These challenges have led to the use of multiple spacecraft to accomplish this mission. The MAV will have to survive extreme environments for extended periods, integrate with the Sample Retrieval Lander (SRL) for support, allow sample tube loading within the payload assembly, prepare for Mars launch, perform Mars ascent, and deliver the payload within specifications – all remotely and from another planet. To accomplish these goals, while working closely with designers on other parts of the campaign, the MAV systems engineering (SE) team has worked to implement emerging systems engineering practices. As a part of this effort, the MAV team became interested and invested in the potential benefits of using Model-Based Systems Engineering (MBSE) practices in the Systems Modeling Language (SysML) alongside traditional systems engineering methods.

NASA has a long history in developing and implementing systems engineering practices. The emergence of MBSE as an improvement upon traditional engineering practice has been challenging to implement within the agency. There has been much research into MBSE but there are few success stories to draw confidence and lessons from within the agency.

Drawing from this base of research, the Mars Ascent Vehicle project became interested in potential benefits of the MBSE paradigm. Some highlighted benefits of a MBSE approach included that MBSE can:

> ‘Increase precision of the system specification and design resulting in reduced downstream errors. Improve

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

traceability between system requirements, design, analysis, and verification information to enhance system design integrity. Improve the ability to maintain and evolve the system specification and design baseline throughout the system lifecycle. Support reuse across projects. Provide a shared understanding of the system to reduce miscommunication among the development team and stakeholders.' [1]

An additional stated benefit, aside from reuse across projects, was the general 'reuse of system specifications and design artifacts.' [2]. While these benefits are attractive, MBSE implementation can require significant up-front investment for later pay off in reducing downstream errors and leveraging of modeled system artifacts. The higher initial investment in systems modeling for MBSE is thought to pay off later, as the act of modeling helps catch early errors: 'The cheapest defect to fix is the one you prevented. And at the heart of this approach is this new kind of engineering artifact called the system model.' [3].

While adopting MBSE does not inherently change the activities a systems engineering team will perform and information contained in deliverables, it is a tool that can change how those deliverables are generated, presented, and reviewed. With a MBSE approach, diagrams and products are generated views of the underlying model and the primary artifact is 'an integrated, coherent, and consistent system model.' [3].

Some agency initiatives such as the MBSE Infusion and Modernization Initiative (MIAMI) [4] and projects such as Europa Clipper [5] helped establish a proving ground for implementation of MBSE within NASA. However, with no NASA systems engineering standards tailored to an MBSE approach, and few projects with success stories to look toward, pursing a solely model-based systems engineering approached carried risks. Notably, the risk that the system model would not be able to generate systems engineering products as traditionally formatted text artifacts for review and approval. To minimize risk to systems engineering product deliverables, in Spring 2020 while in Pre-Phase A development, the MAV SE team decided to pursue a hybrid MBSE approach with traditional SE. In this hybrid approach, MBSE would be pursue and implemented per best practice and the ability of the system model to generate required standard systems engineering products would be demonstrated. However, systems engineering team members would also continue with traditional development of products to ensure successful delivery of desired products. This approach allowed the MAV project to ensure traditionally required deliverables could be produced. This approach also allowed implementation of MBSE practices and thereby later evaluation of benefits and deliverable development capabilities. However, this hybrid effort also meant that some work could be duplicated, or additional work may be required to ensure synchronization of data and product generation.

Since beginning MBSE implementation, the MAV SE team has explored many applications. Some of these applications include development of Concept of Operations (ConOps) diagrams, evaluation of document generation capabilities, ConOps document development, functional decomposition, requirements tracing, physical architecture definition, interface definition and tracing, and Modeling and Simulation (M&S) database trade space exploration.

Section 2 of this paper will discuss MBSE tool selection and environment setup. Following Section 2, this paper will provide more detail to the aforementioned MAV MBSE applications, driving factors, and related findings. These sections will be followed by a general conclusions section.

2. MBSE TOOL SELECTION & SETUP

Before officially pursuing MBSE, the MAV team had to make some decisions. Tools were evaluated, lessons learned referenced, initial use cases identified, and final approval was obtained to begin development of critical systems engineering products in the tool.

MagicDraw [6], a popular SysML editor, was selected as the MAV MBSE tool. This toolset was selected for a few primary reasons. The first reason being it's use at NASA Jet Propulsion Lab (JPL), the primary customer for the MAV team at NASA Marshall Space Flight Center (MSFC). With both teams using MagicDraw, model infrastructure could be more easily managed, and data exchanged. The second reason for the selection of MagicDraw was the ease of access to the tool. At the time, NASA MSFC was actively working on setting up an MBSE environment to provide MagicDraw to any projects wanting to invest in MBSE. This made MagicDraw a cost-free option to the project, with a group already working on setting up and maintaining the software backbone within the agency IT infrastructure, taking any potential direct burden off the project on this front. The third reason for selecting MagicDraw was customizability. While MagicDraw uses standard Unified Modeling Language (UML) and SysML Profiles, these could easily be extended to fit project needs and problems. Additional plugins could be used to extend capabilities of the tool itself. Finally, MagicDraw provides multiple methods for document generation – the key capability necessary to sync model content with document products in the hybrid MBSE and traditional SE approach.

To help with shaping the MAV MBSE effort, lessons learned from the Europa mission studies team, published in their MBSE mission concept study paper [7] and its update [8] were referenced. A few lessons learned from these papers were adapted as establishing part of the MBSE effort:

- 'Automated Web-Based Model Reports are Critical' [8] – this lesson enforced the idea that a web-based platform was necessary for interacting with model information. MagicDraw has a significant learning curve and a web-pashed publishing capability reduces hurdles to consumer engagement with model information. While MAV project was not able to access the same web-based model reporting tool, a similarly capable tool became available from the tool vendor. The MAV project obtained access to MagicDraw Teamwork Cloud via the agency-provided licenses and server environment. Teamwork cloud provided the capability to publish the model from the MagicDraw software environment to an integrated web environment. This web environment allows controlled team member access to model products without the use of licenses and has the optional capability of allowing editing to model data. While this environment is not currently as capable as the tool out of JPL, it does provide a base web-based model reporting capability.

- ‘Keep the Focus on Engineering Products’ [7] – this lesson emphasizes the importance of focusing model work on engineering deliverables. It was decided that in the immediate term, MBSE would be used for ConOps diagrams and ConOps document development with the potential for additional later applications.

- ‘Early Efforts Draw on a Limited Pool of Talent’ [7] – this lesson emphasizes the idea that ‘the first infusions will not have the benefit of an engineering pool with ubiquitous modeling skills’ [7] and that sometimes the best solution is to hire outside expertise rather than training internal talent. Considering the pre-phase A size of the MAV team and the project in general, MAV leadership moved forward with hiring one MBSE subject matter expert.

- ‘Everyone Needs Training, but to Different Levels’ [7] – this lesson pressed the idea that not everyone needs in-depth MBSE training. Different levels of training can be given as needed. The hired MAV MBSE lead trained other systems engineering team members in the tool over time. All team members were exposed to the tool or its products in working sessions and were given in-depth training as needed in areas where the tool could be used to support their work.

In the interest of leveraging traceability benefits offered with a model-based approach, the MAV SE team also carried the goal of later expanding the model to include requirements traceability – showing how requirements were developed and what traces they have to functions, activities, hardware, tests and verification and validation.

Additionally, there was discussion on whether the MAV SE team should approach MBSE strictly from a given methodology [9], such as the Object-Oriented Systems Engineering Method (OOSEM) or a framework such as MagicGrid. Given that the MAV SE team decided to adopt a hybrid MBSE method, letting product deliverables drive the model, no specific methodology or framework was selected. Instead, with a general knowledge base of MBSE methods, modeling best-practices were used for product development.

3. CONOPS DIAGRAM DEVELOPMENT

One systems engineering challenge faced by the MAV team emerged in initial talks with NASA Jet Propulsion Laboratory (JPL) in Fall 2019, as an interface issue – determining commanding/information flow between the MAV and Mars Lander Platform (MLP), the platform that will house the MAV on Mars. In response, the MAV SE team worked to develop information to define the commands needed to trigger MAV activities, internal MAV activities, responses to commanded activities, and order of activities. Functional Flow Block Diagrams (FFBDs) naturally emerged as a tool to capture this information. With MagicDraw available, SysML activity diagrams naturally emerged as a tool to define the FFBDs while also encoding the information within the SysML model. These diagrams also effectively presented this information to team members and stakeholders at both MSFC and JPL.

Given the pre-phase A status of the program, and the potential for design changes, the MAV systems engineering team endeavored to define these functions assuming as little as possible about the technical design of the system. High-level concept space exploration informed development of the diagrams – such as the decision for a two-stage-to-orbit (TSTO) vehicle, and a Solid-Solid Guided-Guided (SSGG) or solid first stage, solid second stage, guided first stage, guided second stage design. To start the definition of functions, the following mission phases were defined for the vehicle: AI&T, Earth Pre-Launch/Cruise/pre-EDL, Mars Surface Operations, Pre-Launch Operations, Launch Day Operations, First Stage Flight, Coast, Second Stage Flight, and OS Payload Delivery. These phases were defined to help logically group vehicle functions. The intent of this was to define mission phases where MAV would have unique sets and series of functions. These phases were created in the MAV model as activities and shown in sequence on an activity diagram, shown in Figure 1.

Each mission phase was then further decomposed by additional activities. Activity diagrams were created for each phase and swim lanes were added to represent MAV subsystems. Activity parameter nodes were used on these lower-level diagrams to represent inputs and outputs to and from the system in communication with the MLP. These inputs, outputs, and functions were defined generally to reduce chance of rework and allow for further specification in later design phases when interface requirements might be established. These activity diagrams succeeded as tools for communicating system functional flow and I/O exchange to all document-based and model-based stakeholders. The success of the activity diagrams was marked by the MAV project adopting them for continual update and tracking concurrent with the system design as it evolves, as a critical SE product. AI&T diagram development ultimately outgrew

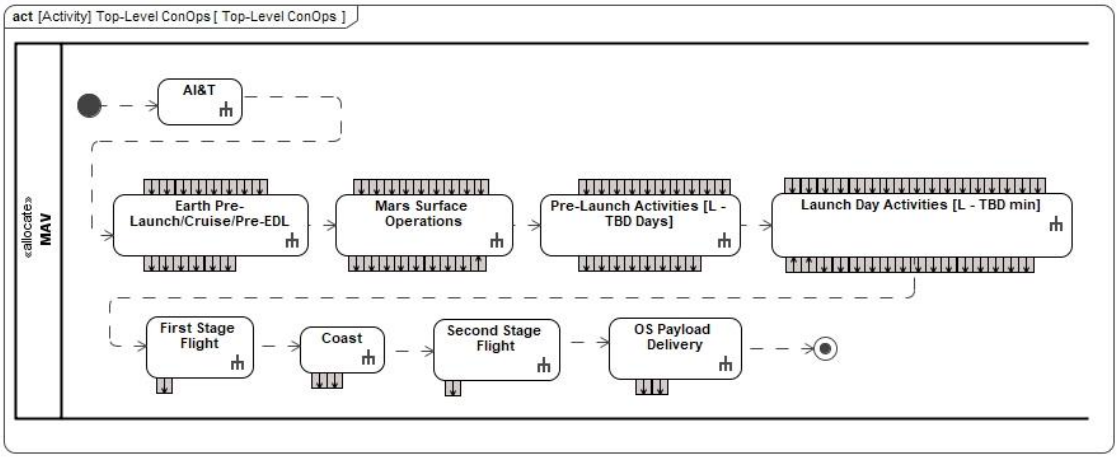


Figure 1: Top – Level ConOps Sequence on Activity Diagram

the ConOps development effort and ownership of the effort was shifted to the AI&T team. The AI&T team have continued development of the AI&T diagrams to capture and sequence activities related to processing, assembly, handling, testing, transport, and delivery of MAV hardware for different MAV vehicle configurations. These diagrams are intended to contribute to MAV AI&T documentation.

4. DOCUMENT GENERATION METHODS

Success of the high-level functional flow implemented by the activity diagrams drove development of the MAV ConOps document. Beyond functional flow and mission phasing, the ConOps is typically a document product addressing stakeholder interests, supporting systems, operational timelines, command & data architecture, communication strategy, integrated logistics support, operational facilities, and contingency & off-nominal operations. At this point, document generation methods were evaluated for publishing sections a ConOps document from the model using the functional flow diagrams and associated information. The MagicDraw document generation methods assessed were Cameo Collaborator publisher (CCP), the MagicDraw Document Modeling Plugin, and MagicDraw Report Wizard. A method that, once configured, could produce a finished, formatted document product was preferred.

Document Generation with Cameo Collaborator Publisher

A CCP document template was created in the model. These templates require the use of the "Collaborator View Diagram", shown in Figure 2.

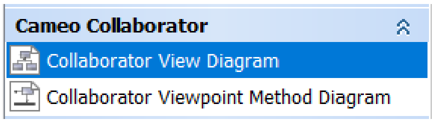


Figure 2: Cameo Collaborator View Diagram

On the Collaborator View Diagram, a Document element is created to establish the document template. View elements are used to compose document sections. Views then expose the desired model packages or elements. Viewpoint elements are then created and exposed by the corresponding view. For this template type, to define the Viewpoint, the "Collaborator Viewpoint Method Diagram", shown in Figure 3, is used.

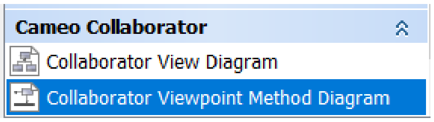


Figure 3: Cameo Collaborator Viewpoint Method Diagram

These Viewpoint method diagrams are activity type diagrams. This diagram offers document modeling specific activities and actions to help with collecting, filtering, sorting, and creating tables. There are also some additional template elements for use such as "paragraph", and "properties table".

Once all document template elements are in place, the template can be used with CCP and pushed to the web environment. These documents can then appear in "model view" where the view sections must be individually selected, or in "document view" where all views are available for scrolling through once one is selected. After generating a test document and adjusting various display settings and options, model data was presented effectively in the web browser. However, this method requires committing model changes, and publishing to teamwork cloud after any edits to the document template or model before it will update. Additionally, the template did not allow the data to be displayed in a traditional MAV document template format with an appropriate cover page, table of contents, table of figures, numbered figures, headers and footers, and page numbering. Also, this document export method only generates the web resource from which a PDF can be printed, not a word document.

Document Generation with MagicDraw Document Generation Plugin

A document generation plugin template was created in the model as well. These templates require the use of the "Views and Viewpoints Diagram", shown in Figure 4.

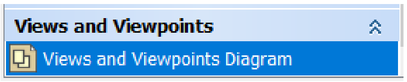


Figure 4: Views and Viewpoints Diagram

On this diagram, a view element is created and can expose desired elements with the expose relationship. Viewpoint templates are included in the diagram palette and require no further definition. Some of these template Viewpoints include table, paragraph, figure, and title page to assist with printing documentation and diagrams. A view conforms with a viewpoint using a conform relationship. Sub-views can be created using view elements and aggregation relationships. In this template, views represent a numbered chapter in the document. The document can be viewed without committing model work to a server using "Document Preview" allowing for efficient adjustment. This document proved to be less customizable. However, this document type does allow DocBook Extensible Markup Language (XML) format export. An XML file was exported, and an XML converter was used to generate a word document. Additionally, an XML editor was used. Results of this effort were closer to a polished document yet required manual rework to adjust formatting.

Document Generation with MagicDraw Report Wizard

MagicDraw report wizard uses word documents populated with text and Velocity Template Language (VTL) to construct a document using model content. Report Wizard includes many pre-defined templates as well as tutorial templates to assist with learning how to use VTL with MagicDraw. Due to using a standard word document as the template base, this template was by far the most customizable and conformed to the traditional template desired for application as a MAV standard. These templates could be more complex, automatically pulling types of information and diagrams out of the model in a given order. Additionally, they could be simpler, outputting only a requested diagram or table where applicable.

5. CONOPS DOCUMENT GENERATION TEMPLATE DEVELOPMENT

To ensure seamless integration with traditional systems engineering deliverables, a ConOps document template was created using MagicDraw Report Wizard. The implementation of document generation for any alternative typically involves a model-development process in parallel to the development of the primary SE content represented by the SysML model of the system of interest. For the selected document generation alternative of the VTL script in the report wizard, the development effort would focus on authoring the script in concert with iterative tailoring of the model so that the two could work together to product the stakeholder-desired results. In Spring 2020, the original template was developed and consisted of a standard cover page, signature page, table of contents, table of figures, sections for each mission phase diagram followed by a listing for each action on the diagram and an acronyms table. Using the mission phase activity diagrams in the model, the template automatically created a document header for each phase, printed the appropriate diagram and the name of each action below the diagram.

Over time, the diagrams were further refined to accommodate high-level design changes, such as to an unguided upper stage. Actions were evaluated for uniqueness and updated to activities if an action represented the same thing in two different diagrams. Documentation was then added to these activities to clearly define what the activity represented. A similar approach was taken with activity parameter nodes; the nodes were typed with signals or blocks if representing the same thing in multiple locations. Documentation was added to the parameter node types. Using this method, generic tables were used to generate separate lists of the activities, parameter nodes (inputs and outputs), and events on the diagram for each mission phase. Activity and parameter node type documentation was displayed, ensuring consistent documentation regardless of diagram table line. The document template was updated to print these tables following each diagram in the document.

Following the initial development effort, it was clear in the generated document that the same text for a given activity was repeated in different sections of the document. To reduce repetition in the document, the division of these activities became the focus of the document structure. Mission phase diagrams for stowed through launch day diagrams were printed in a first section. Following this section, a new section was created to list only activities present in all stowed phases (all phases prior to ejection from MLP on day of launch – while MAV is still stowed within the MLP) and in the day of launch phase. After this section, a section was created for stowed and day of launch phase specific sequences. A matrix was created in this section to show which unique activities applied to what mission phase diagrams. This matrix was followed by a table listing these unique activities. To follow this section, another was created to include unique launch day specific sequences and functions. This section included a table listing the unique launch day activities with their documentation and an additional column for designation of a given activity as being specific to final countdown before launch from Mars. Following this document section, another section was created to list activities shared between any ascent phases or the launch day phase. A matrix was created in MagicDraw to show which activities applied to what phases. The VTL template was iteratively modified to target this matrix and follow it with a table listing the activities and their documentation. After this section, additional sections were created for each ascent phase (first stage flight, coast, second stage flight, and payload delivery) with their respective mission phase diagram and activity description table.

All diagrams, matrices, and tables were generated within the model for use by the VTL script. The final VTL script targeted the revised model artifacts and compiled most of the document sections. As the output of the process was a word document, stakeholders could then add in additional ConOps information as desired per traditional processes.

6. FUNCTIONAL DECOMPOSITION DEVELOPMENT

Following the first iteration of ConOps diagrams, in Summer 2020, the MAV SE team turned its focus toward early requirements development. At this point, a detailed vehicle-focused functional decomposition was developed per the practice established with ConOps – that is, in addition to the activity diagrams describing functional flow for the ConOps document. Following how the ConOps diagrams were divided by mission phase, those phases were used as a base to help define MAV vehicle level functions. While the ConOps FFBDs included several MAV functions, they served primarily to specify the orchestration of MAV with the MLP, rather than focus on the functional hierarchy of the MAV itself. The ConOps functions were recognized as valuable inputs but were not structured in the manner of a hierarchical functional decomposition and in some cases included details below the system level. Working up from the ConOps diagrams with respect to each mission phase, parent functions were defined at the vehicle level. A vehicle-level function stereotype, a requirement generalization, was created within the model to define these elements. This function stereotype was created due to the desire to refrain from labeling these artifacts as requirements and presenting them as such.

SysML Requirements diagrams were used initially to create and build up this vehicle level functional decomposition in a block diagram format. However, MagicDraw relation maps within emerged as a way to display the full functional decomposition. The vehicle-level functional decomposition in a relation map became a valuable and effective resource for discussing the functions with team members and stakeholders. After repeated refinement of these functions, those duplicated in multiple phases were evaluated for uniqueness. While some were unique, duplicates that were not unique were deleted and a single representative function was associated with the appropriate mission phases. Later, repeated functions between phases were evaluated for uniqueness and if they were the same, they were used as a single element in the model. Figure 5 shows parent function definition in a relation map.

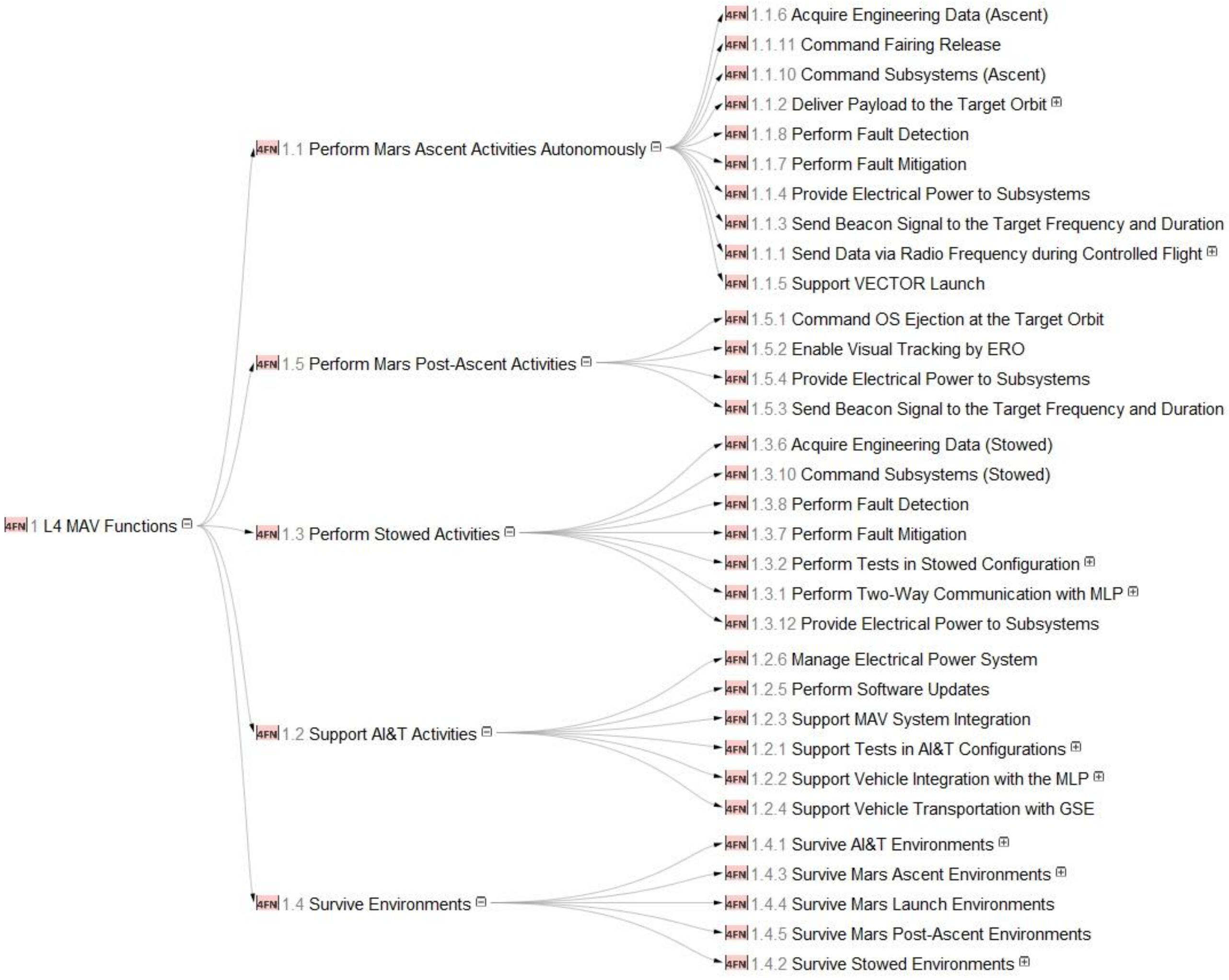


Figure 5: Functional Decomposition in Relation Map

Once completed, the functions in this vehicle-level functional decomposition were used to draft vehicle level functional requirements in IBM Rational Doors Next Generation (DNG) [10].

7. REQUIREMENTS TRACEABILITY

While requirements can be managed in MagicDraw, given the successful implementation on previous programs and to ensure compatibility with JPL requirements artifacts, in Spring 2020 DNG was selected as the MAV requirements management tool. Using the MAV functional decomposition from MagicDraw, requirements were written in DNG – the MAV requirements source of truth. One goal for MagicDraw was to build requirements traceability including links between the functions in MagicDraw and the functional requirements derived from them, defined in DNG. After a brief evaluation of available tools, Cameo DataHub was selected as a primary candidate due to its ready availability as an integrated MagicDraw product. Custom MAV Requirement stereotypes were developed in the model to represent different requirement levels and properties were defined to match data fields from DNG. In Spring 2021, after matching requirement data fields, DataHub was used to link MagicDraw to DNG and a one-way schema was developed to sync requirement metadata from DNG to MagicDraw. Requirements were then synced from DNG to MagicDraw on an as needed or weekly basis.

After the functional requirements in DNG were drafted from the functions in MagicDraw, DataHub was used to import the functional requirements set to MagicDraw. With the draft functional requirements in MagicDraw, a Derive Requirement Matrix diagram was created in MagicDraw to create formal traces between the vehicle level functions and draft functional requirements. This matrix was then used to identify any gaps or orphans in the derivation process, i.e., functions that did not have a derived requirement or functional requirements that did not trace back to a function. If a draft functional requirement was identified without a trace to a function, the requirement source was identified. Identifying the source either led to the realization that a function as missing from the functional decomposition, or the functional decomposition was not the driving force behind a requirement. In the case the functional decomposition and MAV functions were not the requirement driver, the identified requirement source was used to reclassify the requirement as a type of requirement other than functional. Identifying these non-functional requirements in the functional requirements set and shifting them into another set ensured requirements were being more rigorously assessed by the appropriate requirement source owners based on the original requirement drivers. This process proved very useful for identifying gaps in vehicle level functional requirement derivation traces, ensuring there was a driving source behind each requirement. These traces served as a tool to refine and inform the final documented set of requirements.

Once the traces between the functions and functional requirements were established and refined using the matrix, a requirements derivation map was used to visualize these traces for presentation and discussion. With the relation criteria set to include the functions hierarchy, the functional decomposition was shown in the same manner as on the original functional decomposition relation map. However, with the additional relation criteria “Derived”, the derived functional requirements from DNG were shown downstream of the appropriate parent function on the diagram.

The draft requirements set continued to grow with interface requirements, design requirements, environmental requirements, and performance requirements. As the requirements set grew in DNG, it became more difficult to identify types of requirements or requirements derived from a particular resource in MagicDraw. At this point, a data attribute was added in DNG to designate the requirement owner – a MAV subsystem representative, and the requirement source such as interface diagrams, functions, and particular documents. With this requirement attribute imported into MagicDraw, requirements became easier to sort and filter. This ability to filter requirements by tagged source allowed for more efficient creation of traces within the model to source elements.

With DNG serving as the MAV requirements source of truth, methods to generate documentation from the requirements set were evaluated. Using MagicDraw to generate the system requirements document was considered. However, with DNG serving as the requirements source of truth, a native DNG document generation template was developed. MagicDraw then primarily served to hold additional requirements traceability details for requirements definition, evaluation, and reference purposes.

8. PHYSICAL ARCHITECTURE DEFINITION

While refining ConOps and functional decomposition elements, it became clear that a high-level physical architecture could be defined in preparation for functional allocation activities. In Spring 2021, an initial block definition diagram was created to begin defining MAV system blocks. These blocks were initially defined using a block definition diagram and were intended to describe overall composition and naming convention for MAV hardware and assembly definition. Blocks were kept general enough not to specify specific hardware solutions, but still satisfy functions and functional requirements. With a physical architecture modeled, those blocks could be used to support traces between requirements, functions, and activities.

The SysML block definition diagram (BDD) was useful for initially defining the blocks and relationships. However, as the physical architecture grew, it became more difficult to effectively display in full. Since block level definition was sufficient at this stage of development, a relation map was created to show the block hierarchy. The highest levels of

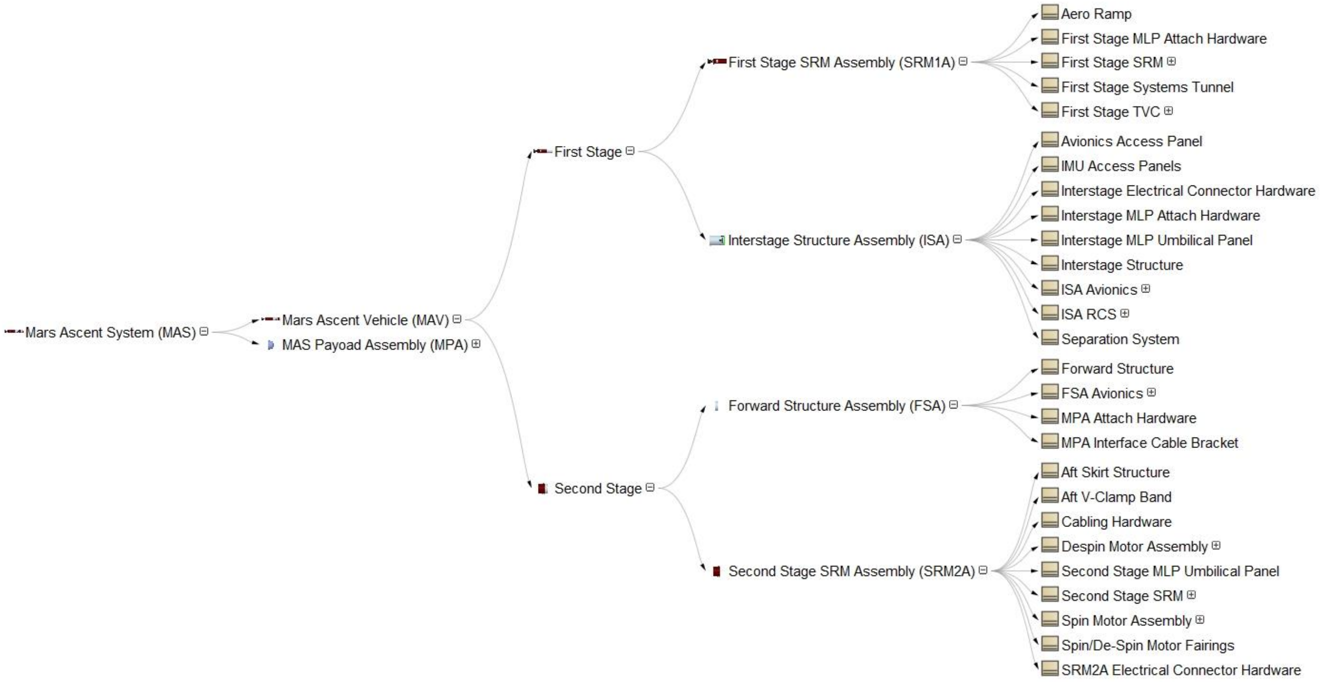


Figure 6: Physical Architecture in Relation Map

MAV definition in the relation map are shown in the following figure, Figure 6.

Due to positive reception and continued use of the MAV physical architecture relation map, a physical architecture document was drafted for use as a reference document in the MAV System Requirements Review (SRR). To allow for appropriate text sizing within image exports of the physical architecture map, the architecture was export as a set of diagrams. First, a map showing the breakdown from the MAV system context level to the main system blocks was included. Following this diagram, additional diagrams were generated to show the hardware breakdown within each main system blocks with additional levels included as necessary for diagram clarity. Using the MAV word document template, the document was populated and lines of VTL were included in the relevant locations to print the current diagrams from the model when generated. Once completed, the document was pushed through MagicDraw Report Wizard and the relevant diagrams were printed in the applicable locations.

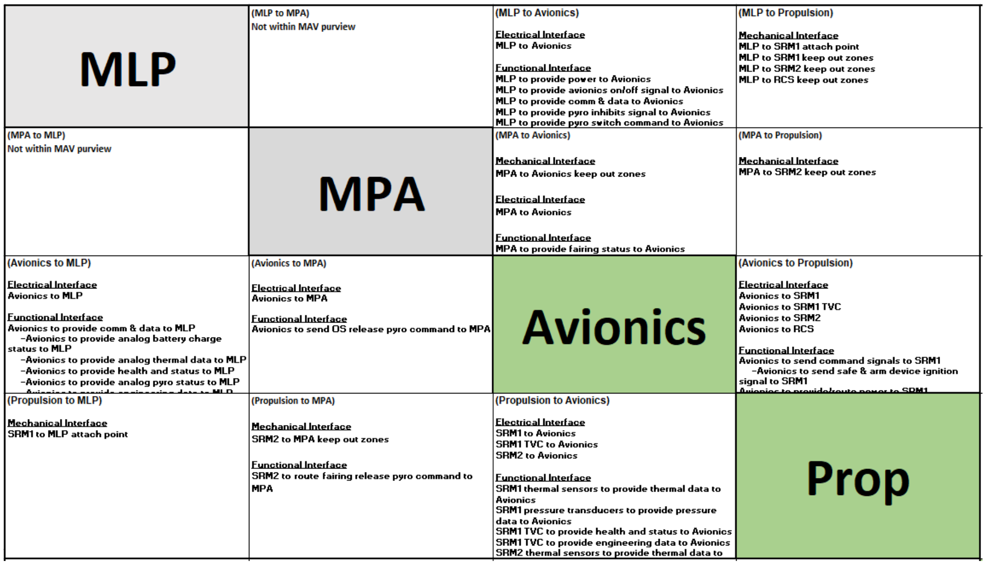
The intent of the physical architecture is not to serve as an authoritative source for system design information or interface definition, as the SysML satisfy relationship is an assertion to be later demonstrated in the verification process, and a detailed physical solution can be added later by means of the SysML allocate relationship or generalization, depending on modeling practice. However, the rough physical architecture provides further logical organization of the functionality and provides initial traceability for functional allocation. Specific quantities and names of hardware were not reflected at this level of definition. The physical architecture is intended to be updated as needed and on a periodic basis to coincide with major milestones or design updates.

9. INTERFACE DEFINITION

As interface requirement development began in Fall 2020, an N-squared (N2) diagram was used to assist in initial definition of functional and physical interface items. This interface diagram was developed at the system level by placing primary MAV systems along a matrix diagonal and stepping through each potential subsystem interface to identify functional and physical interface items. This N2 diagram was drafted and refined in an excel sheet. Once the N2 diagram reached a stable draft state, vehicle level interface requirements were drafted in DNG using the N2 diagram defined interfaces. A portion of the initial N2 diagram can be seen in Figure 7.

In Summer 2021, after the interface requirements were drafted in DNG, additional updates and changes to interface content were continuously incorporated. Due to the lack of formal tracing between the N2 diagram defined interfaces and draft interface requirements, interface information between the two sources quickly became inconsistent. To better maintain and capture the interface requirements and their sources, the N2 diagram interfaces were defined and modeled in MagicDraw and formal traces were created to the interface requirements in DNG.

MagicDraw does not have a built-in N2 diagram but does have Internal Block Diagrams (IBDs). IBDs capture similar information to N2 diagrams in a different format including data, structural, power, and fluid exchange between systems. These diagrams were used to capture data, power, and



MLP	(MLP to MPA) Not within MAV purview	(MLP to Avionics) Electrical Interface MLP to Avionics Functional Interface MLP to provide power to Avionics MLP to provide avionics on/off signal to Avionics MLP to provide comm & data to Avionics MLP to provide pyro inhibits signal to Avionics MLP to provide pyro switch command to Avionics	(MLP to Propulsion) Mechanical Interface MLP to SRM1 attach point MLP to SRM1 keep out zones MLP to SRM2 keep out zones MLP to RCS keep out zones
(MPA to MLP) Not within MAV purview	MPA	(MPA to Avionics) Mechanical Interface MPA to Avionics keep out zones Electrical Interface MPA to Avionics Functional Interface MPA to provide fairing status to Avionics	(MPA to Propulsion) Mechanical Interface MPA to SRM2 keep out zones
(Avionics to MLP) Electrical Interface Avionics to MLP Functional Interface Avionics to provide comm & data to MLP -Avionics to provide analog battery charge status to MLP -Avionics to provide analog thermal data to MLP -Avionics to provide health and status to MLP -Avionics to provide analog pyro status to MLP	(Avionics to MPA) Electrical Interface Avionics to MPA Functional Interface Avionics to send OS release pyro command to MPA	Avionics	(Avionics to Propulsion) Electrical Interface Avionics to SRM1 Avionics to SRM1 TVC Avionics to SRM2 Avionics to RCS Functional Interface Avionics to send command signals to SRM1 -Avionics to send safe & arm device ignition signal to SRM1
(Propulsion to MLP) Mechanical Interface SRM1 to MLP attach point	(Propulsion to MPA) Mechanical Interface SRM2 to MPA keep out zones Functional Interface SRM2 to route fairing release pyro command to MPA	(Propulsion to Avionics) Electrical Interface SRM1 to Avionics SRM1 TVC to Avionics SRM2 to Avionics Functional Interface SRM1 thermal sensors to provide thermal data to Avionics SRM1 pressure transducers to provide pressure data to Avionics SRM1 TVC to provide health and status to Avionics SRM1 TVC to provide engineering data to Avionics SRM2 thermal sensors to provide thermal data to	Prop

Figure 7: Portion of Initial MAV N2 Diagram

command interface flows between different MAV systems. Using the MAV physical architecture, IBDs were generated for MAV blocks at various levels. N2 diagram information from excel was processed and sorted in excel to aid in data analysis and entry. By using on key words in interface definition strings, columns for originating system, direction, interface item (commands, data, etc.), and receiving system were generated.

Interface BDDs were developed for subsystems with interface definitions. On these BDDs, existing block composition was displayed. New blocks were created for any hardware specified in the N2 diagram that was not previously defined in the model. An interface BDD was also created for housing and defining interface blocks. By looking at the list of interface items from the N2 excel data, a list of interface blocks was generated: Data, Electrical, Mechanical, Keep Out, Lifting and Handling, Attach Point, Propellant Loading, Umbilical Cutout, and Alignment Cutout. For interfaces block with interface item flow, flow properties were defined (data, power, propellant). On the subsystem interface BDDs, proxy ports were added to subsystems for each type of data input and output. These proxy ports were typed with the relevant interface block to represent the port.

On the system IBDs, parts with defined interfaces from the N2 diagrams were shown with their ports. Ports were conjugated as necessary to allow reverse property flow on the supplier side. Connectors were defined between ports to represent specific interface item flows (power, commands, thermal data, pressure data, etc.). Aspects were applied and used to color code based on port type (data, electrical, mechanical). From these IBDs, Blackbox and Whitebox ICD table diagrams were generated in MagicDraw to display interface information in tabular format. These IBDs were then exported and shared with MAV team members to facilitate sharing and discussion of interface information. A high-level avionics to propulsion electrical IBD is shown in Figure 8. A high-level portion of a generated Whitebox ICD table is shown in Figure 9.

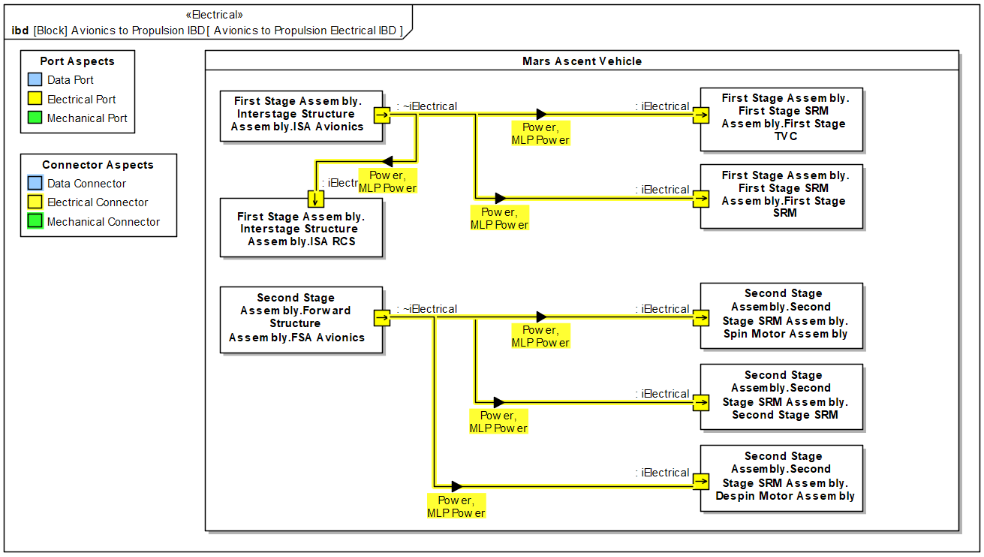
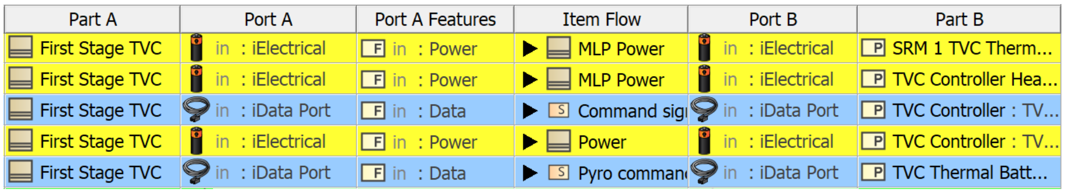


Figure 8: Draft MAV Interface Diagrams



Part A	Port A	Port A Features	Item Flow	Port B	Part B
First Stage TVC	in : iElectrical	F in : Power	▶ MLP Power	in : iElectrical	P SRM 1 TVC Therm...
First Stage TVC	in : iElectrical	F in : Power	▶ MLP Power	in : iElectrical	P TVC Controller Hea...
First Stage TVC	in : iData Port	F in : Data	▶ S Command sigı	in : iData Port	P TVC Controller : TV...
First Stage TVC	in : iElectrical	F in : Power	▶ Power	in : iElectrical	P TVC Controller : TV...
First Stage TVC	in : iData Port	F in : Data	▶ S Pyro comman	in : iData Port	P TVC Thermal Batt...

Figure 9: Auto-Generated Whitebox ICD Table

To assess the emerging gaps between interface requirements and N2 information, now in MagicDraw, a matrix was created in the model. Ports, the representation of interface hardware, were generated along one matrix axis and the interface requirements from DNG were placed along the second matrix axis. Using this matrix, the draft interface requirements in DNG were traced to the satisfying interface hardware. After completing this trace, gaps in the trace were completed to assess requirements that did not trace to interface hardware and interface hardware that did not trace to a requirement. Interface requirements that did not trace to interface hardware implied that either the requirement was not interface derived, or that an interface was missing from the interface diagrams. Interface hardware without an interface requirement implied that either the interface was no longer relevant, or that an interface requirement was missing. These gaps were discussed, and the interface requirements and hardware set were adjusted to close interface to requirement traceability.

10. M&S DATABASE CONCEPT DEVELOPMENT

In Summer 2021, during the initial MAV requirements development process, and in preparation for SRR, MagicDraw was assessed as a potential database tool for M&S management. Per NASA standard 7009 [11], model and simulation metadata for critical models is to be recorded and managed to provide decision makers with enough information to understand model results and applications. While there are established M&S database management tools within NASA, MagicDraw was evaluated as a potential candidate to house M&S data due to its use in developing other systems engineering information and products on the MAV project. Additionally, this implementation of MagicDraw could potentially provide a no cost added M&S database tool for the project.

The MAV M&S leads identified a base set of requirements for a MAV M&S management tool:

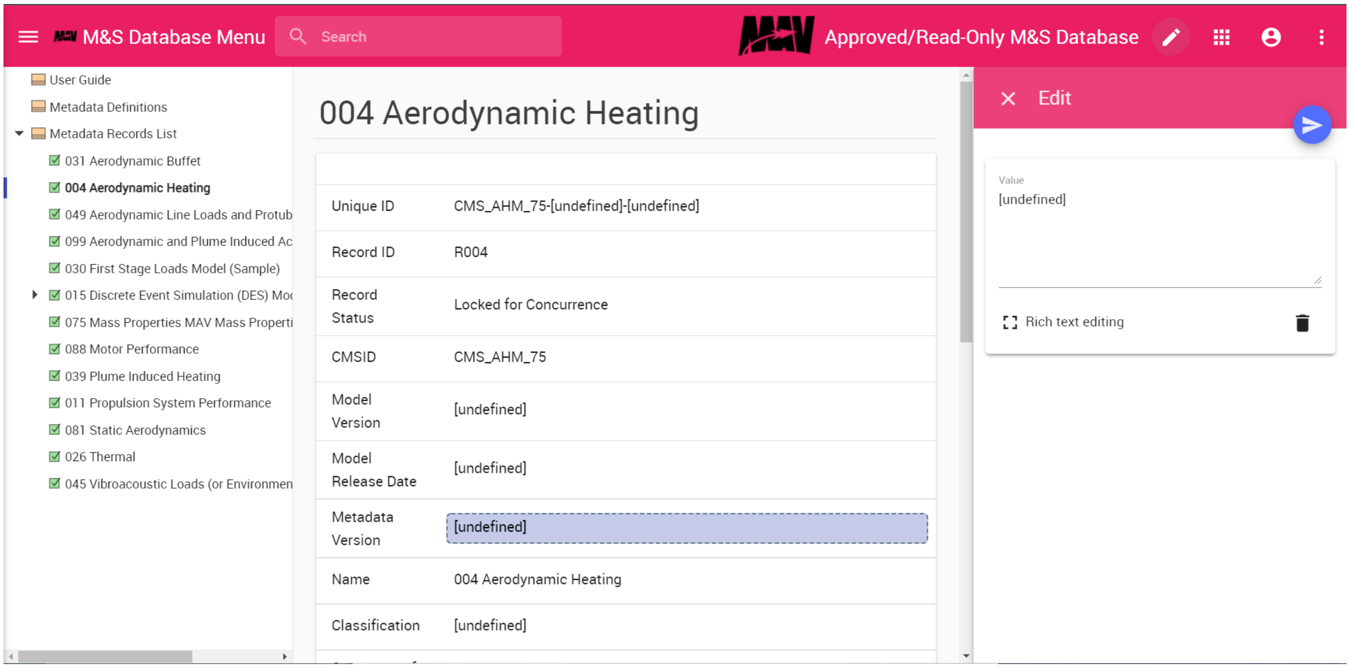
1. The ability to house all standard metadata items.
2. A user-friendly format that allowed user data entry, editing, commenting, and review.
3. A method of approving metadata records.
4. The ability to apply appropriate security markings.
5. A baselining capability with report generation.
6. The ability for management to easily access and search models and metadata.

Using these base drivers, a concept M&S database in MagicDraw was developed. While CCP did not present information in a traditional document format, it did present information from the model in a tailorable and consumable format through a web platform. This capability was ideal for implementing the database due to its connection with the model and because users do not need MagicDraw software to access, edit, search, and browse published resources. Additionally, the MagicDraw Teamwork Cloud (TWC) web environment allows for strict control of users and access permissions. Per NASA access forms, users are only granted access upon request. Once granted access to TWC, users are individually given access to appropriate resources and specific levels including the ability to review, edit, or manage content.

One benefit of implementing the M&S database in MagicDraw was that M&S records would be modeled as elements that could be traced to requirements and verification elements. However, the environment would need to be proved as effective and capable of managing M&S records at the base standard of other available options. A Requirement Verification Model stereotype, a block generalization, was created within the model for use in defining M&S model elements and their metadata. Attributes were created on this Requirement Verification Model stereotype to accommodate the necessary metadata information fields.

A CCP template was developed to publish three views. One view for a user-guide page for database navigation instructions and a downloadable PowerPoint slide of instructions for distribution and a second view for M&S metadata definitions. The third view included the list of M&S records included in the database linked to sub-views containing the models with their metadata description tables. In each model metadata description view, an additional table was included to list previous versions of metadata for the same model – listed as a sub-view to the individual model views in the model. These metadata tables were generated within their respective views in the CCP template using the properties table predefined template activity. The properties mode was set to standard and the standard properties menu for the Requirement Verification Model elements was customized to show the desired metadata items. Using this method, the properties tables published were editable in the TWC browser. CCP templates do allow for custom built tables using provided table structure elements however, these diagrams did not publish with editable values – so the properties table method was used instead. However, while the properties table method published editable values, if there was previously no value for a particular item, the value field would publish empty and would not be editable. A custom derived property field was later added to the Requirement

Verification Models for configuration management purposes. This custom derived property was called unique ID and combined three different strings from three metadata identification properties into a unique ID. A visual of the TWC M&S database concept user interface is shown in Figure 10.



004 Aerodynamic Heating

Unique ID	CMS_AHM_75-[undefined]-[undefined]
Record ID	R004
Record Status	Locked for Concurrence
CMSID	CMS_AHM_75
Model Version	[undefined]
Model Release Date	[undefined]
Metadata Version	[undefined]
Name	004 Aerodynamic Heating
Classification	[undefined]

Value
[undefined]

Figure 10: M&S Database User Interface in TWC

Once the CCP database template was completed, to update the database resources, a M&S representative can push an update to the resource in MagicDraw using CCP. Once published, any new Requirement Verification Models from MagicDraw automatically appears in the published database with any associated sub-views. To create a new Requirement Verification Model record, a model developer requests a new item listing. Then, the M&S representative creates a requirement verification model element in the M&S database package in MagicDraw, populates registration metadata, and ensures data entry fields contained an ‘[undefined]’ note. After the model was committed and published, assigned model developers can then populate the undefined metadata fields. Metadata record editing permissions for developers are removed once a record moves to review. After review, the record is fully locked and placed in the final M&S database resource. To export viewable model metadata records, a TWC user can generate a quick PDF report from the web resource. However, as noted previously, this report lacks special document formatting.

Given the limited use of TWC PDF reports, a VTL document template was created to generate a polished M&S database report. The M&S database VTL template iteratively loops through all requirement verification models owned by a package in a selected scope and prints the model metadata information into a formatted word document. The document is populated with general front matter intended to be reused between baselined versions. This ensures that the polished document can be generated from the database within seconds and require minimal adjustment before being handed over – saving the modeling and simulation team the additional time that would normally be warranted for this kind of database to document information capture task. This document serves as the hardcopy of model metadata for baselining purposes within MAV configuration management. While the modeled database itself is not serving as the configuration managed baseline, the ability to generate the required documentation automatically from the managed database has proven useful.

This M&S database proof of concept successfully met base capability expectations and implementation of the tool will be tested with metadata record information available at and after the MAV SRR in Fall 2021.

11. CONCLUSIONS & FORWARD WORK

With support from agency initiatives and project leadership, MBSE tools have been used in the MAV systems engineering process to develop and deliver official systems engineering products – in a traditional document-based format, along with model-based products. A high-level timeline of MAV MBSE implementation is shown in Figure 11. During initial implementation, in pre-phase A project lifecycle development, the MAV team invested in MBSE expertise to guide MBSE implementation and introduce MBSE concepts to MAV team members. Targeted MBSE benefits included increased system specification and design quality, improved SE artifact traceability, increased reuse of specification and design information, increased consistency of information with model-generated products, and enhanced communication of developed system information. Access to

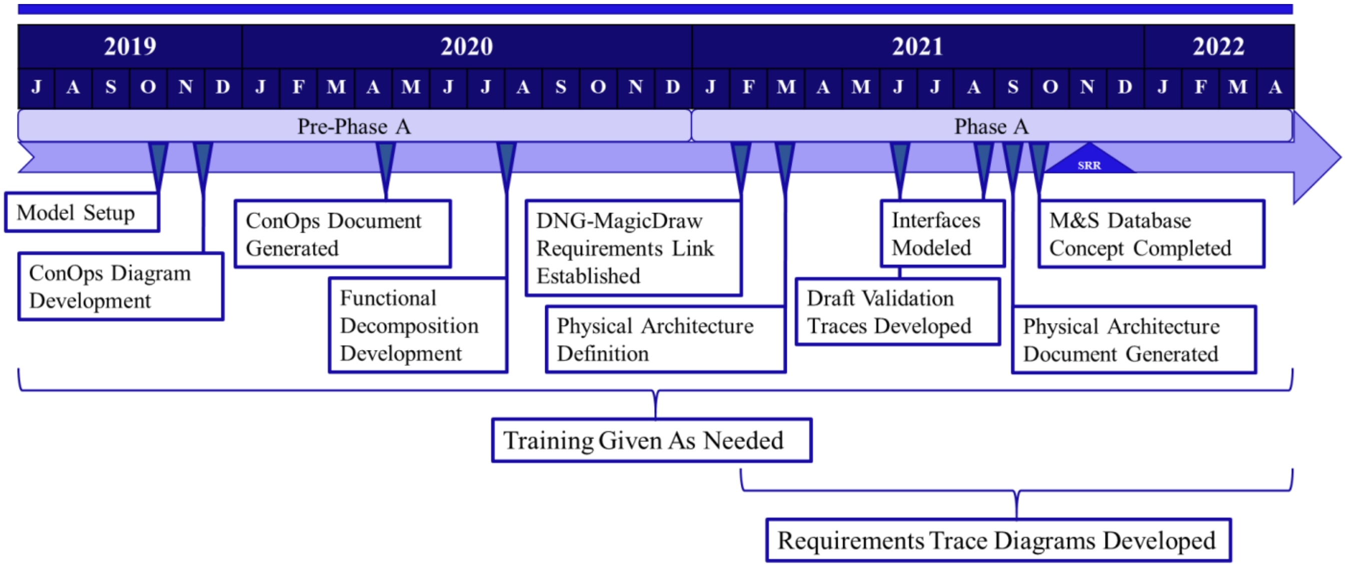


Figure 11: Timeline of High-Level MAV MBSE Implementation

an MBSE tool was acquired and MBSE practices were applied to MAV systems engineering artifact development.

Since development of the MAV MBSE model, some of these benefits have been assessed to some extent. The MAV MBSE model has been used to reduce gaps between requirements definition and design information – increasing system specification quality. Model elements representing system interface, functions, requirements, and physical architecture have been traced together to assess completeness of design specification and development threads – increasing SE artifact traceability. While the entirety of these threads was not reflected in the initial requirements document, TWC was used to publish detailed information, as available, for team viewing. In the 2 years since initial ConOps diagram and document generation template development, ConOps diagrams, functions, and templates have been adjusted and reused as high-level design changes emerged from design analysis cycles. With the implementation of MBSE artifact development for the ConOps document, consistency of unique model elements across views and generated document resources from the model was demonstrated. With the development of visualized interface diagrams using N2 diagram developed interface data, interface information was shared with team members with success as an enhanced method of viewing. Using CCP and TWC for M&S database development and establishment, the ability of MBSE to contain system design information and share it with desired team members was visualized and will be further assessed.

The MAV SRR in Fall 2021 provided the MAV systems engineering team the opportunity to present their model-based work. Following questions on systems engineering process and requirements traceability, the MAV systems engineering model was shared at the MAV SRR. Using various diagrams mentioned in this paper, the MAV SE team walked through the model and showed many of the MBSE applications used to develop requirements, functions, ConOps, interfaces, physical architecture, and other system information. Using the model, a great breadth and variety of information, as well as how it connected, was efficiently shared. The model proved to be a welcome source of detailed information and the project's use of MBSE was noted as a strength that gave the review board confidence in handling of requirements flow-down.

Throughout implementing MBSE on MAV, a general pattern emerged when considering a new application/model usage:

1. Identify modeling methods for application.
2. Identify relations to existing model information and weigh potential benefit.
3. Consider method/process for sharing information contained in the potential application.
4. Consider how to address baselining/configuration management and security needs.
5. Assess issues for the implementation and refine application details.
6. Obtain necessary approvals to implement modeling application.
7. Implement modeling application and continue to refine as necessary.

Given this general pattern, steps were learned on how to better formulate and plan for model applications. Oftentimes, a combination of the TWC web portal publishing capability and document generation using VTL templates was necessary to answer MAV systems engineering use cases: TWC for general sharing of information and collaboration and VTL templates to generate documents for baselining.

In addition to the MBSE applications discussed in this paper, the MAV SE team has assessed applicability of MBSE for validation item development, MAV human factors (HF) team artifact integration, and MAV flight software (FSW) team artifact integration.

As part of an early-project lifecycle verification and validation (V&V) effort, the MAV MBSE model was used to trace the MAV functional decomposition and ConOps functions to satisfying hardware, validation objectives, tests contributing to validation, validation deficiencies, risks associated with validation deficiencies and potential flight test objectives. In this effort, existing SE model information and relationships were leveraged with model-based practices to increase traceability of validation item development and allowed for systematic, efficient relationship development to populate traces. As the MAV requirements development effort matures, a final V&V approach will be developed.

The MAV SE team worked with the MAV HF team to discuss integration of HF task analysis and assessment items and development of human factors products from the system model. This potential integration with the MAV HF team would allow human factors requirements sources, task analysis functions, specification, and compliance artifacts to be traced to MAV systems architecture and requirements within the model. Additionally, HF products could potentially be generated from the model-based environment.

Additionally, the MAV SE team has worked with the MAV FSW team to provide a FSW modeling sandbox area within the MAV model. The MAV FSW team MBSE integration area introduces the potential for draft FSW algorithms, signals, and requirements to be included in MAV system model architecture traceability and design definition.

In addition to these areas, the MAV SE team intends to assess the ability of interface information developed and traced within the system model to interface control document development. The MAV MBSE effort is ongoing and system model applications are continuing to be developed and assessed.

APPENDIX: ACRONYMS

AI&T	Assembly, Integration, and Test
BDD	Block Definition Diagram
CCP	Cameo Collaborator Publisher
ConOps	Concept of Operations
DNG	Doors Next Generation
ERO	Earth Return Orbiter
FFBD	Functional Flow Block Diagram
FSW	Flight Software
HF	Human Factors
IBD	Internal Block Diagram
JPL	Jet Propulsion Lab
MAS	Mars Ascent System
MAV	Mars Ascent Vehicle
M&S	Model and Simulation
MBSE	Model-Based Systems Engineering
MLP	Mars Lander Platform
MPA	MAS Payload Assembly
MSFC	Marshall Space Flight Center
MSR	Mars Sample Return
N2	N-squared
NASA	National Aeronautics & Space Administration
OCE	Office of the Chief Engineer
OS	Orbiting Sample
RCS	Reaction Control System
SE	Systems Engineering
SRC	System Requirements Cycle
SRL	Sample Retrieval Lander
SRM	Solid Rocket Motor
SRR	System Requirements Review
SSGG	Solid-Solid Guided-Guided
SSGU	Solid-Solid Guided-Unguided
SysML	Systems Modeling Language
TSTO	Two Stage to Orbit
TVC	Thrust Vector Control
TWC	Teamwork Cloud
UML	Unified Modeling Language
V&V	Verification & Validation
VTL	Velocity Template Language
XML	Extensible Markup Language

ACKNOWLEDGEMENTS

The author would like to thank MAV leadership and the MAV SE team for their ongoing support of MBSE application. Special thanks to SE leads Peter Ma and Darius Yaghoubi for their role in guiding the adaptation of MBSE to traditional SE process, Shawn Maynor for contributions to ConOps diagram development, MAV MBSE intern Richard Simmang for implementing the interface modeling, Thomas Wickline for his role in tailoring DNG to improve MagicDraw connectivity, interface leads Megan Beattie and John Crowley for informing model-based application of interface work, MAV MBSE interns Liam Durbin, Scott Nealon, and the MAV AI&T team for adapting the MagicDraw AI&T diagrams. Thanks to the verification and validation effort team, HF team, FSW team for supporting discussion of MBSE application concepts, and the entire MAV team for their warm reception of MBSE resources and products. Finally, the author would like to thank the NASA OCE MBSE Architecture Team for providing requested capabilities and solutions.

REFERENCES

[1] "Architecting Spacecraft with SysML, A Model-Based Systems Engineering Approach", by Sanford Friedenthal and Christopher Oster. Friedenthal & Oster, 2017.

[2] A Practical Guide to SysML, "The Systems Modeling Language", Third Edition by Sanford Friedenthal, Alan Moore, and Rick Steiner. Morgan Kaufmann, 2015.

[3] SysML Distilled, "A brief guide to the systems modeling language", by Lenny Delligatti with forewords by Rick Steiner and Richard Soley. Pearson Education, 2014.

[4] J. B. Holladay, J. Knizhnik, K. J. Weiland, A. Stein, T. Sanders and P. Schwindt, "MBSE Infusion and Modernization Initiative (MIAMI): "Hot" Benefits for Real NASA Applications," 2019 IEEE Aerospace Conference, 2019, pp. 1-14, doi: 10.1109/AERO.2019.8741795.

[5] T. Bayer, "Is MBSE helping? Measuring value on Europa Clipper," 2018 IEEE Aerospace Conference, 2018, pp. 1-13, doi: 10.1109/AERO.2018.8396379.

[6] MagicDraw Web site: https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/

[7] T. J. Bayer et al., "Model Based Systems Engineering on the Europa mission concept study," 2012 IEEE Aerospace Conference, 2012, pp. 1-18, doi: 10.1109/AERO.2012.6187337.

[8] T. Bayer et al., "Update on the Model Based Systems Engineering on the Europa Mission Concept Study," 2013 IEEE Aerospace Conference, 2013, pp. 1-13, doi: 10.1109/AERO.2013.6496855.

[9] Estefan, Jeff. (2008). "Survey of Model-Based Systems Engineering (MBSE) Methodologies". INCOSE MBSE Focus Group.

[10] IBM Doors Next Generation Web site: https://www.ibm.com/products/requirements-management-doors-next

[11] NASA. 2016. NASA-STD-7009 REV A Change 1 – Standard Models and Simulations.

BIOGRAPHY

Isabeta Rountree *received a B.S. in Industrial & Systems Engineering from the University of Alabama in Huntsville in Spring 2020. She has been studying Model-Based Systems Engineering since Spring 2019, when she became an undergraduate research assistant in the Complex Systems Integration Lab at UAH. She was the MAV MBSE intern before Spring 2020 when she joined Geocent, now Sev1Tech, a Jacobs Space Exploration Group teammate at Marshall Space Flight Center as the full time MAV MBSE lead. She has since continued her work implementing MBSE as a part of the Mars Ascent Vehicle systems engineering team.*