NASA TECHNICAL HANDBOOK

Office of the NASA Chief Engineer

NASA-HDBK-1009A
Approved: 2025-03-12

METRIC/SI (ENGLISH)

NASA SYSTEMS MODELING HANDBOOK FOR SYSTEMS ENGINEERING

DOCUMENT HISTORY LOG

Status	Document Revision	Change Number	Approval Date	Description
Baseline			2022-11-14	Initial Release
Revision	A		2025-03-12	Key changes: 1. Expanded the scope from modeling for ConOps, Requirements, and V&V products to modeling for the following NASA SE Work Products: Stakeholder identification and Expectations definition, ConOps, MOE definition, Technical Requirements, MOP, TPM definition, V&V Requirements, Planning, Results and Reports. 2. Added Appendix F – ConOps Template with Model content (following the SE Handbook template) 3. Expanded the metamodel in Section 7 for: Stakeholder Expectations, MOE, MOP, TPM, and V&V. 4. Expanded Section 8 to provide example diagrams and tables to support metamodel expansion (Stakeholder Expectations, MOE, MOP, TPM, and V&V examples) 5. Expanded Section 9 to include diagram and table examples for the expanded list of SE Work Products. Expanded the content in the ConOps, V&V, and Requirements views; Added Stakeholder identification and expectation definition, MOE definition, MOP, and TPM views.

FOREWORD

This NASA Technical Handbook is published by the National Aeronautics and Space Administration (NASA) to provide engineering information; lessons learned; possible options to address technical issues; classification of similar items, materials, or processes; interpretative direction and techniques; and any other type of guidance information that may help the Government or its contractors in the design, construction, selection, management, support, or operation of systems, products, processes, or services.

This Handbook establishes how system modeling using Systems Modeling Language™ (SysML®) can be integrated with the NASA Systems Engineering (SE) processes in NPR 7123.1, NASA Systems Engineering Processes and Requirements. The systems engineering products covered in this Handbook are those generated from the following NASA SE Engine processes: Stakeholder Expectation Definition, Technical Requirements Definition, Product Verification, and Product Validation. The work products covered are Stakeholder identification and Expectations definition, Concept of Operations (ConOps), Measures of Effectiveness (MOE) definition, Technical Requirements, Measures of Performance (MOP), Technical Performance Measures (TPM) definition, Verification & Validation (V&V) Requirements, Planning, Results and Reports.

This Handbook contains sections on model planning, setting up the model including model organization, the metamodel used to demonstrate the system modeling elements and relationships, a section on model building that provides example SysML® models following the metamodel, and a section on generating diagrams and tables from the system model to support the NASA SE work products.

Submit requests for information via "Email Feedback" at https://standards.nasa.gov. Submit requests for changes to this Handbook via Marshall Space Flight Center (MSFC) Form 4657, Change Request for a NASA Engineering Standard, or the "Suggest a Change to this Standard" link on the Standard's Summary Page at https://standards.nasa.gov.

Tracy Osborne signed for	2025-03-12
Joseph W. Pellicciotti NASA Chief Engineer	Approval Date

TABLE OF CONTENTS

NASA SYSTEMS MODELING HANDBOOK FOR SYSTEMS ENGINEERING

1. SCOPE

1.1 Purpose

This NASA Technical Handbook shows how system modeling using Systems Modeling Language™ (SysML®) can be integrated with the NASA Systems Engineering (SE) processes in NPR 7123.1, NASA Systems Engineering Processes and Requirements. The modeling covered in this Handbook support systems engineering work products generated from the following NASA SE Engine processes: Stakeholder Expectation Definition, Technical Requirements Definition, Product Verification, and Product Validation. For a list of the specific work products covered see section 4.1.1 of this Handbook. The content of this version includes these four processes based on a survey conducted through the NASA Agency Model Based System Engineering (MBSE) Community of Practice.

This Handbook contains sections on model planning, setting up the model including model organization, the metamodel used to demonstrate the system modeling elements and relationships, model building that provides SysML® model examples, and generating diagrams and tables from the system model to support NASA SE work products.

The system modeling method in this Handbook is tool-agnostic. The Handbook references SysML v1.7; however, any modeling language can be applied to the metamodel in section 7 to depict the NASA SE elements and relationships.

The modeling approach in this Handbook leverages NASA modeling practices but does not reflect all NASA modeling methods. If readers have their own modeling approach, they can use the metamodel as a reference to trace back to their modeling approach.

> A companion model to this Handbook is available on the NASA Technical Standards Website at https://standards.nasa.gov/standard/NASA/NASA-HDBK-1009[1].

1.2 Applicability

1.2.1 This Handbook is applicable to system modelers using Object Management Group® (OMG®) SysML® version 1.7. These modelers include individuals who have varying levels of experience with the SysML® modeling language and knowledge of how systems engineering is conducted at NASA, which should include the efficient and effective application of NPR 7123.1 and NASA/SP-2016-6105, NASA Systems Engineering Handbook.

[1] The companion model was built using MagicDraw; however, the modeling method can be used with different modeling tools. Note: usage does not constitute an official endorsement, either expressed or implied, by NASA.

1.2.2 This Handbook is approved for use by NASA Headquarters and NASA Centers and Facilities. This language applies to the Jet Propulsion Laboratory (a Federally Funded Research and Development Center), other contractors, recipients of grants, cooperative agreements, or other agreements only to the extent specified or referenced in the applicable contracts, grants, or agreements.

1.2.3 References to "this Handbook" refer to NASA-HDBK-1009A; references to external documents state the specific document information.

1.2.4 In this Handbook, the terms "may" or "can" denote discretionary privilege or permission, "should" denotes a good practice and is recommended but not required, "will" denotes expected outcome, and "is/are" denotes descriptive material or a statement of fact.

1.2.5 This Handbook, or portions thereof, may be referenced in contract, program, and other Agency documents for guidance.

2. REFERENCE DOCUMENTS

2.1 General

Documents listed in this section provide references supporting the guidance in this Handbook. Latest issuances of referenced documents apply unless specific versions are designated. Access reference documents at https://standards.nasa.gov or obtain documents directly from the Standards Developing Body, other document distributors, information provided or linked, or by contacting the office of primary responsibility designee for this Handbook.

2.2 Government Documents

NASA

NPR 7123.1, NASA Systems Engineering Processes and Requirements

NASA-STD-7009, Standard for Models and Simulations

NASA-HDBK-7009, NASA Handbook for Models and Simulations: An Implementation Guide for NASA-STD-7009

NASA/SP-2016-6105, NASA Systems Engineering Handbook

Model-Based System Engineering, NEN (https://nen.nasa.gov/web/mbse)

2.3 Non-Government Documents

American Institute of Aeronautics and Astronautics

Parrott, E., and Weiland, K. (2017). "Using Model-Based Systems Engineering to Provide Artifacts for NASA Project Life-Cycle and Technical Reviews," AIAA SPACE and Astronautics Forum and Exposition. (https://doi.org/10.2514/6.2017-5299)

International Council on Systems Engineering

INCOSE - International Council on Systems Engineering. (n.d.). Retrieved October 4, 2022. "INCOSE Initiatives". INCOSE. (https://www.incose.org/incose-member-resources/initiatives)
Morkevicius, A.; Aleksandraviciene, A.; Mazeika, D.; Bisikirskiene, L.; & Strolia, Z. (2017).

"MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems." INCOSE International Symposium (Vol. 27, No. 1, pp. 136-150). (https://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2017.00350.x)

Object-Oriented SE Method Working Group, INCOSE (https://www.incose.org/communities/working-groups-initiatives/object-oriented-se-method) (OOSEM Process Baseline (1/2020))

Stevens Institute of Technology Systems Engineering Research Center

SEBoK Editorial Board. (2022). "The Guide to the Systems Engineering Body of Knowledge (SEBoK)," v. 2.6, R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed 9/6/2022. (www.sebokwiki.org).

Other

Friedenthal, S.; Moore, A.; and Steiner, R. (2014). "A Practical Guide to SysML: The Systems Modeling Language," 3rd ed. Boston: Morgan Kaufmann.

ISO/IEC 19514: 2017(E), Information Technology – Object Management Group Systems Modeling Language (OMG SysML®)

Karban, R.; Crawford, A.G.; Trancho, G.; Zamparelli, M.; Herzig, S.; Gomes, I.; Piette, M.; Brower, E. (2018). "The OpenSE Cookbook: A Practical, Recipe Based Collection of Patterns, Procedures, and Best Practices for Executable Systems Engineering for the Thirty Meter Telescope." (https://trs.jpl.nasa.gov/handle/2014/48358)

Object Management Group (OMG). (2024). "System Modeling Language (SysML), Version 1.7." (https://sysml.org/sysml-specs/)

Object Management Group (OMG). (2022). "What is SysML?" OMG SysML. (https://www.omgsysml.org/what-is-sysml.htm)

2.4 Order of Precedence

2.4.1 The guidance established in this Handbook does not supersede or waive existing guidance found in other Agency documentation.

2.4.2 Conflicts between this Handbook and other documents will be resolved by the delegated Technical Authority.

3. ACRONYMS, ABBREVIATIONS, AND DEFINITIONS

See Appendix H.

4. MODEL-BASED SYSTEMS ENGINEERING (MBSE) OVERVIEW

The purpose of this Handbook is to show how system modeling using SysML® can be integrated with the NASA systems engineering (SE) processes in NPR 7123.1, NASA Systems Engineering Processes and Requirements. This section will provide background information about NASA's systems engineering processes and system modeling.

4.1 NASA Systems Engineering Process Overview

NPR 7123.1 provides a generic description of systems engineering as it is applied throughout NASA. There are three sets of common technical processes in NPR 7123.1: system design, product realization, and technical management. The processes in each set and their interactions and flows are illustrated in **Figure 4.1-1**, **NASA Systems Engineering Engine**. NASA SE utilizes artifacts (example: ConOps Report, Requirements Specifications, and V&V Plans) that are inputs to and outputs from these common technical processes. For more information on the NASA SE Engine and the 17 SE common technical processes, refer to NASA/SP-2016-6105, Section 2.1. A description of each of the common technical processes is captured in Appendix A of this Handbook.

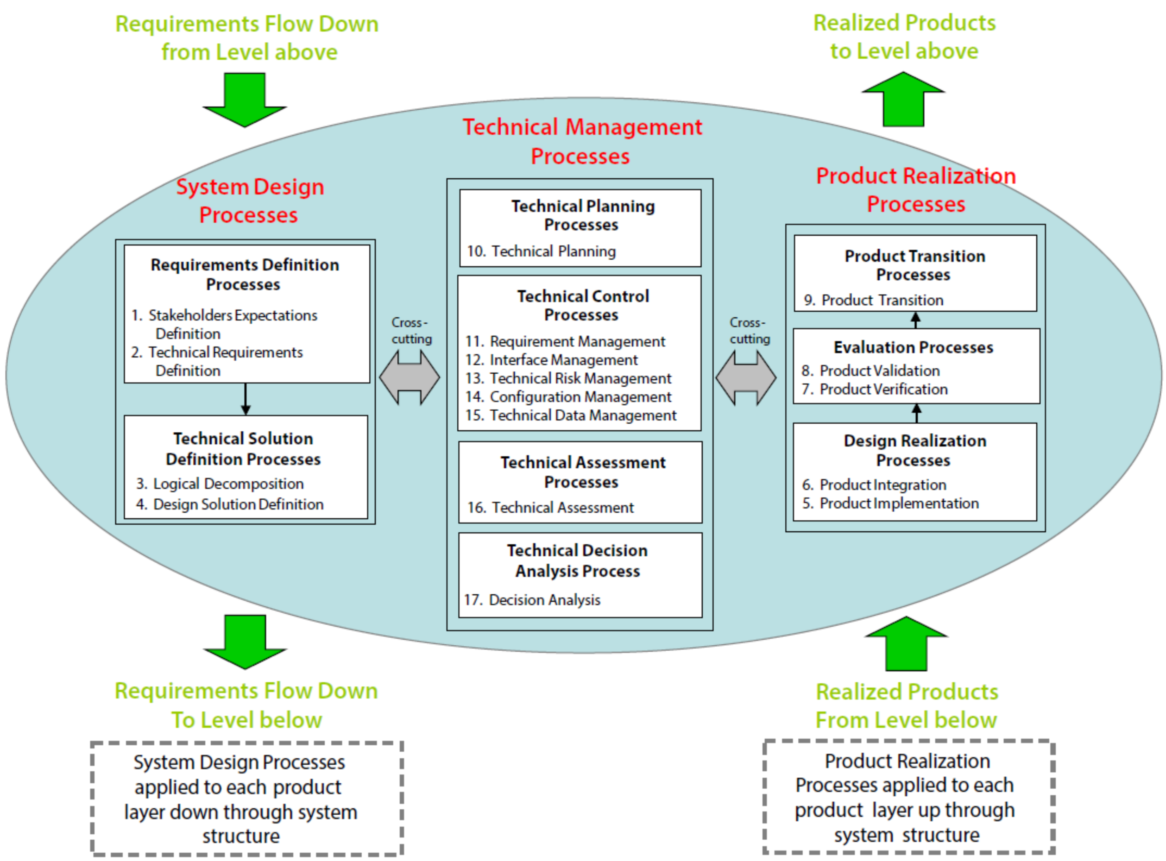


Figure 4.1-1 NASA Systems Engineering Engine[2]

[2] NPR 7123.1D, Figure 3-1

4.1.1 SE Processes vs. Products

This section will discuss the differences between NASA systems engineering processes detailed in **Figure 4.1-1**, **NASA Systems Engineering Engine**, versus the products these processes generate.

The NASA SE Engine and the 17 common technical processes are the means by which we do Systems Engineering, and the products are what is generated as a result of the process. For example, a list of NASA SE primary work products is detailed in NPR 7123.1, Table 5-1.

The following is a list of NASA SE products covered in this handbook for each of the processes within the scope.

Table 4.1-1 NASA SE Processes and Products Covered in Handbook

NASA SE Common Technical Processes	List of Products Covered in the Handbook
Stakeholder Expectation Definition	• Stakeholder identification and Expectations definition • Concept of Operations (ConOps) • Measures of Effectiveness (MOE) definition
Technical Requirements Definition	• Technical Requirements • Measures of Performance (MOP) • Technical Performance Measures (TPM) definition
Product Verification	• Verification Requirements • Verification Planning • Verification Results and Reports
Product Validation	• Validation Requirements • Validation Planning • Validation Results and Reports

4.2 MBSE and the NASA Systems Engineering Process

The International Council on Systems Engineering (INCOSE) has defined MBSE as follows: "MBSE is the formalized application of modeling to support system requirements, design, analysis, V&V activities beginning in the conceptual design phase and continuing throughout development and later life-cycle phases."[3]

In terms of the NASA SE Engine, MBSE supports the common technical SE processes by using system models to capture the definitions and relationships of the system of interest. From the system models, diagrams and tables (including matrices) are used to implement the SE processes, and SE work products are generated to support technical reviews for programs and projects.

[3] INCOSE - International Council on Systems Engineering. (n.d.). Retrieved October 4, 2022. "INCOSE Initiatives". INCOSE. (https://www.incose.org/incose-member-resources/initiatives)

4.3 Three Aspects of MBSE

MBSE has three aspects: the modeling language, the modeling methodology, and the modeling framework. These are described in detail in the following subsections.

4.3.1 Modeling Language

An implicit requirement to author a model is a modeling language, much like how programming utilizes a programming language and human communication utilizes a natural language to represent concepts and pass information. The modeling language facilitates the description of the system of interest using graphical constructs. INCOSE recognizes the SysML® modeling language for specifying, analyzing, designing, and verifying complex systems. This Handbook uses SysML® as the modeling language.

4.3.1.1 SysML® Diagram Types

SysML® has nine diagram types (see **Figure 4.3-1**, **SysML® Diagrams**). There are four behavior diagrams: activity diagram (act), sequence diagram (sd), state machine diagram (stm), and use case diagram (uc). There is a requirement diagram (req) that captures requirement hierarchies and relationships. There are four types of structure diagrams: block definition diagram (bdd), internal block diagram (ibd), package diagram (pkg), and parametric diagram (par).[4]

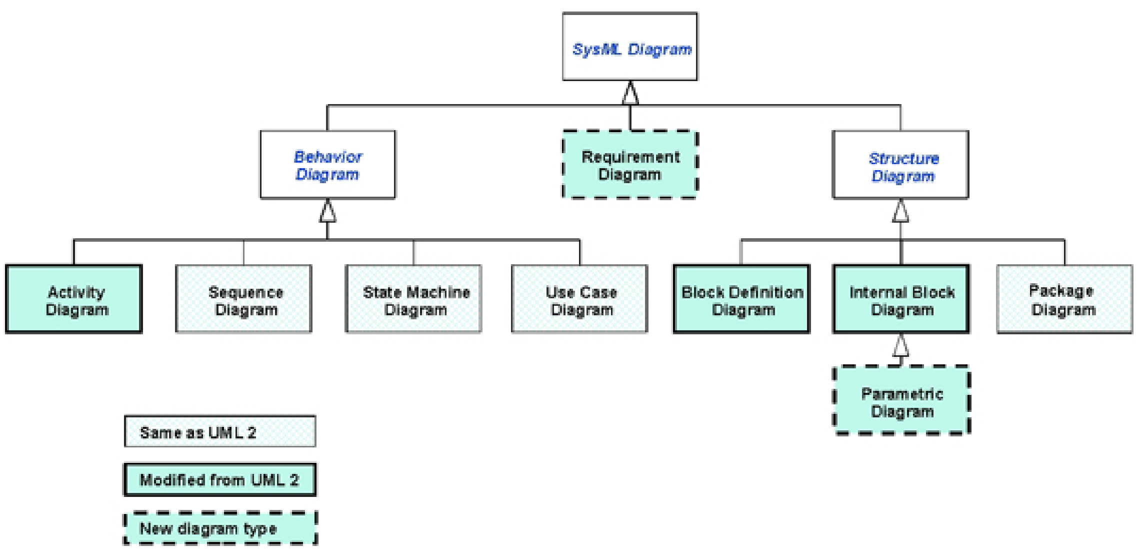


Figure 4.3-1 SysML® Diagrams[5]

Note: **Figure 4.3-1** highlights the nine SysML diagrams and how they compare to Unified Modeling Language (UML 2), a graphical language developed for software. SysML represents a subset of UML 2 with some extensions[5].

[4] ISO/IEC 19514: 2017(E)
[5] Object Management Group (OMG). (2022). "What is SysML?" OMG SysML. (https://www.omgsysml.org/what-is-sysml.htm)

4.3.1.2 Modeling Pillars of SysML®

SysML® diagrams are often grouped within four modeling pillars: structure, behavior, requirements, and parametrics (see **Figure 4.3-2**, **Four Pillars of SysML®**). Each pillar supports the common SE activities used to define a system in a model to develop an SE product. The structure pillar supports the depiction of logical and physical layers such as systems, subsystems, components, and interfaces. The behavior pillar supports domains like system functionality, system interactions, system response, and system information and data flow. The requirements pillar supports specifications and V&V. The parametric pillar supports constraints and mathematical statements. Together, the pillars build a collective context across the entire SysML® model, integrating model elements and diagrams to support SE product generation.

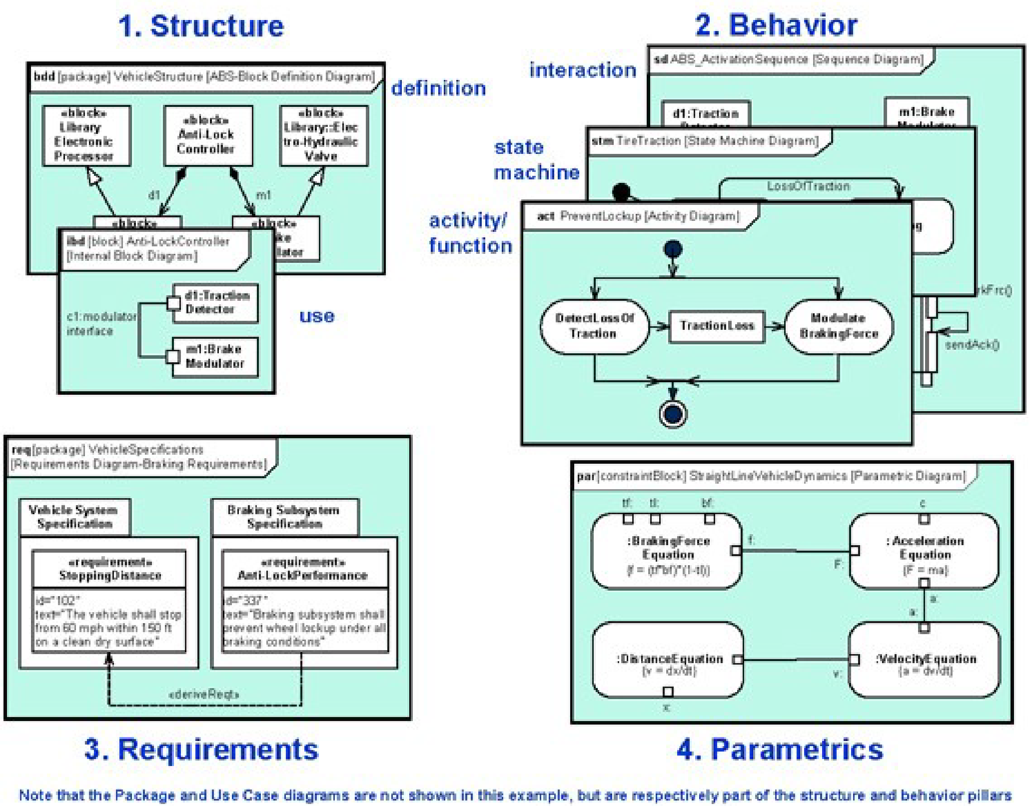


Figure 4.3-2 Four Pillars of SysML®[6]

4.3.2 Modeling Methodology

While modeling languages like SysML® provide enhanced structure and rigor to SE constructs for capturing information in the model, the step-by-step processes to build a model and to support data output are not provided by the language. A modeling methodology provides a structured approach for building and using models.

[6] Object Management Group (OMG). (2022). "What is SysML?" OMG SysML. (https://www.omgsysml.org/what-is-sysml.htm)

The modeling methodology in this Handbook follows the NASA SE Engine with additional model-specific steps not included in the NASA SE Engine: "Model Planning" and "Setting Up" the Model. These model-specific steps are detailed in sections 5 and 6. These steps were leveraged from an INCOSE standard called Object-Oriented Systems Engineering Method (OOSEM) which details a systems engineering process, similar to the NASA SE process, along with model-specific steps (see Appendix B for more information on OOSEM).

Figure 4.3-3, **NASA SE Engine with Modeling Specific Steps**, shows "Model Planning" occurring in Technical Process 10, Technical Planning, and "Setting Up the Model" occurring in the start of the System Design Process as Step 0.

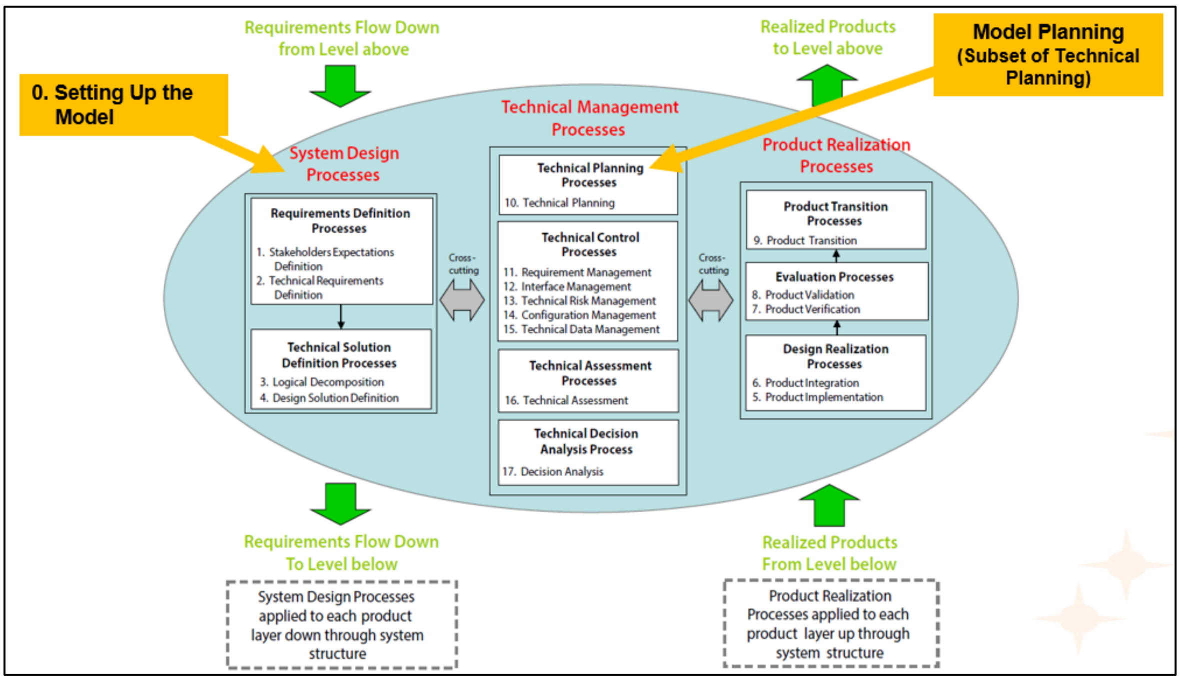


Figure 4.3-3 NASA SE Engine with Modeling Specific Steps

For the Modeling Methodology in this handbook, the objective is to develop models that align with the NASA SE process and support SE work-product development. Each of these processes are further described in NPR 7123.1 and the NASA SE Handbook. For example, the System Design Process steps from the NASA SE Handbook is captured in **Figure 4.3-4**, **System Design Process Interactions and Flows**.

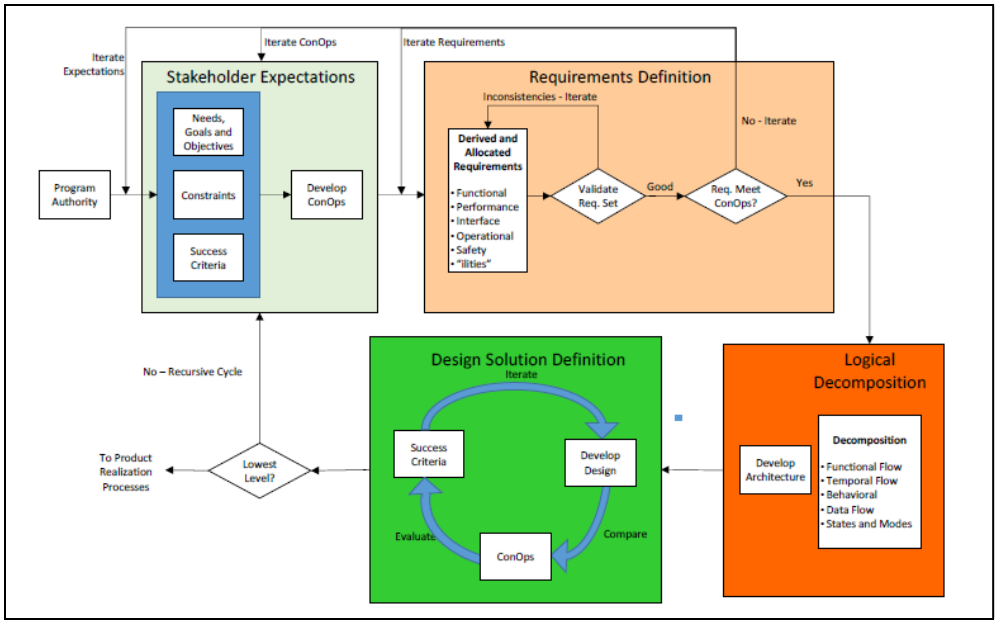


Figure 4.3-4 System Design Process Interactions and Flows[7]

4.3.3 Modeling Framework

A modeling framework provides the approach to organizing the system elements and relationships within the model.

The modeling framework in this Handbook leverages the MBSE Grid framework and tailors it to the NASA SE Engine (see Appendix B for more information on MBSE Grid).

The modeling framework for this handbook relates the Modeling Pillars of SysML® to the NASA Systems Engineering Technical Processes shown in **Figure 4.3-5**, **Processes 1-9 in the NASA SE Engine**.

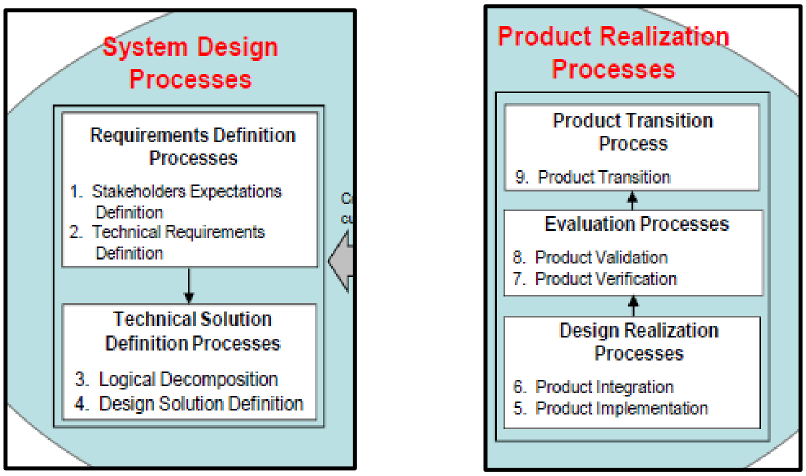


Figure 4.3-5 Processes 1-9 in the NASA SE Engine

[7] NASA/SP-2016-6105, Revision 2, Figure 4.0-1

The grid is leveraged to depict the NASA Systems Engineering Technical Processes in rows and the Modeling Pillars of SysML® in columns. Note that the grid shows only technical processes 1-9 and does not include the Technical Management Processes 10 thru 17. **Figure 4.3-6**, **Grid with NASA SE Processes and SysML Pillars**, shows the MBSE Grid with NASA SE Processes depicted in the rows and the 4 SysML pillars represented in columns.

		SYSML PILLAR			
	TECHNICAL PROCESSES	REQUIREMENTS	BEHAVIOR	STRUCTURE	PARAMETERS
NASA SYSTEMS ENGINEERING TECHNICAL PROCESSES (1-9)	1. Stakeholder Expectations Definition				
	2. Technical Requirements Definition 3. Logical Decomposition				
	4. Design Solution Definition				
	5. Product Implementation 6. Product Integration				
	7. Product Validation 8. Product Verification				
	9. Product Transition				

Figure 4.3-6 Grid with NASA SE Processes and SysML Pillars

A metamodel can be used to relate modeling elements and relationship across all 4 SysML pillars that create the content for the system model's diagrams and tables. Section 7 describes the metamodel for NASA SE elements and relationships described for this handbook.

The grid can be used to organize SysML diagrams and tables from the model within the cross-sections. Appendix C shows a grid from **Figure 4.3-6** with the diagrams and tables from this handbook populated in the cross-sections.

5. MODEL PLANNING

Model planning provides the technical details about the modeling activities and what products can be expected from the models. In the NASA SE Engine, model planning occurs in the Technical Planning Process (see Technical Process 10 in **Figure 4.1-1**, **NASA Systems Engineering Engine**). The modeling plan is a technical plan that is a subset of the Systems Engineering Management Plan (SEMP). The SEMP documents how NASA systems engineering requirements and practices of NPR 7123.1 will be addressed throughout the project/program life cycle. The modeling plan documents how modeling will support those system engineering requirements and practices throughout the project/program life cycle. This plan includes a list of project products that can be supported by the system models, modeling resources for the project, roles and responsibilities, modeling tools, modeling conventions, model accessibility, and model management for the project/program.

The modeling plan is established early in the life cycle. As the system matures and progresses through the life cycle, the modeling plan should be updated as necessary to reflect the current environment and resources. Sample modeling plans are available on the NASA MBSE Community of Practice website at https://nen.nasa.gov/web/mbse/ (Search: Modeling Plan Template).

6. SETTING UP THE MODEL

Setting up the model includes establishing modeling conventions, standards, and model organization. As described in section 4.3.2, setting up the model occurs in the beginning of the System Design Processes (see **Figure 4.3-3**, **NASA SE Engine with Modeling Specific Steps**).

Modeling conventions include establishing naming conventions for model element and package names. Modeling standards include establishing standard profiles and other modeling standards based off the needs of the project/program.[8] Model organization refers to the package structure and hierarchy setup for capturing the system model. Organizing the model provides a standard package structure that best reflects the system hierarchy.[8] A sample model organization that relates to the NASA SE Engine is depicted in **Figure 6-1**, **Sample Model Organization**. Projects/programs can select a model organization that best fits their needs.

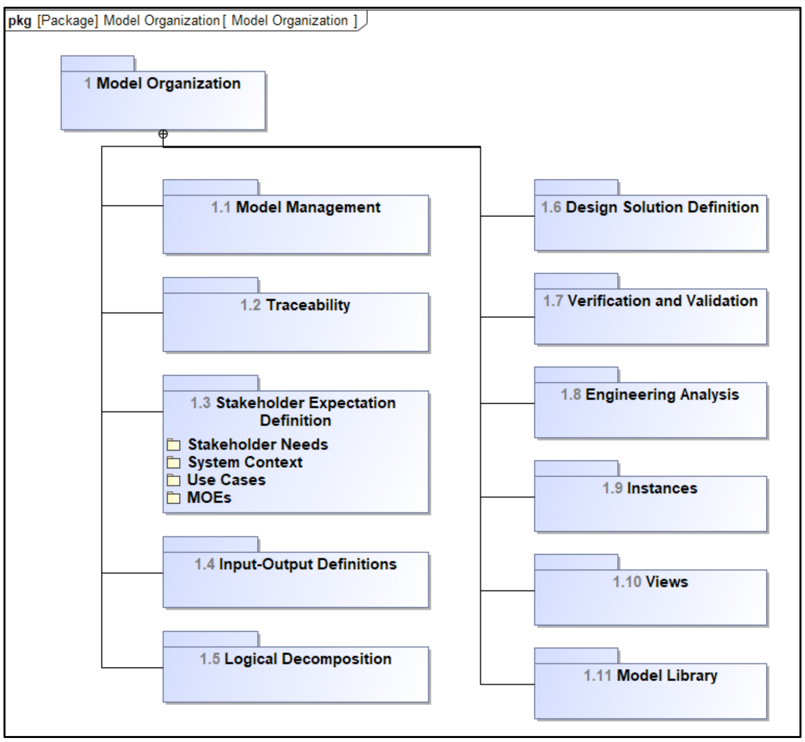


Figure 6-1 Sample Model Organization

The results of establishing modeling convention, metamodel, modeling standards, and model organization are documented in the Modeling Plan.

[8] Friedenthal, S.; Moore, A.; and Steiner, R. (2014). "A Practical Guide to SysML: The Systems Modeling Language," 3rd ed. Boston: Morgan Kaufmann.

7. THE METAMODEL

A metamodel is a depiction of the system modeling elements and their relationships. **Figure 7-1**, **Metamodel Based on NASA Systems Engineering (SE) Elements and Relationships** and **Figure 7-2**, **V&V Metamodel Based on NASA Systems Engineering (SE) Elements and Relationships**, show the metamodel for system modeling based on NASA SE elements and relationships described in NPR 7123.1. In the metamodel, [ ] are used to capture the SysML® language-specific element or relationship type (e.g., requirement, block, activity, refines, derives).

Notes:

- In the metamodel, [ ] are used to capture the SysML® language-specific element or relationship type, however any other modeling language may be substituted in the [ ].
    - This handbook references System Modeling Language (SysML), Version 1.7.[9]
    - This document and the metamodel will be updated to reflect any changes needed in the future, for example SysML version 2.0.

If readers have their own modeling approach, they can use the metamodel to trace back to their modeling approach for Stakeholder Expectations Definition, Technical Requirements Definition, and V&V. The metamodel and any assumptions should be documented in the modeling plan for a given project/program (see section 5, Model Planning, for more details).

[9] Object Management Group (OMG). (2024). "System Modeling Language (SysML), Version 1.7." (https://sysml.org/sysml-specs/)

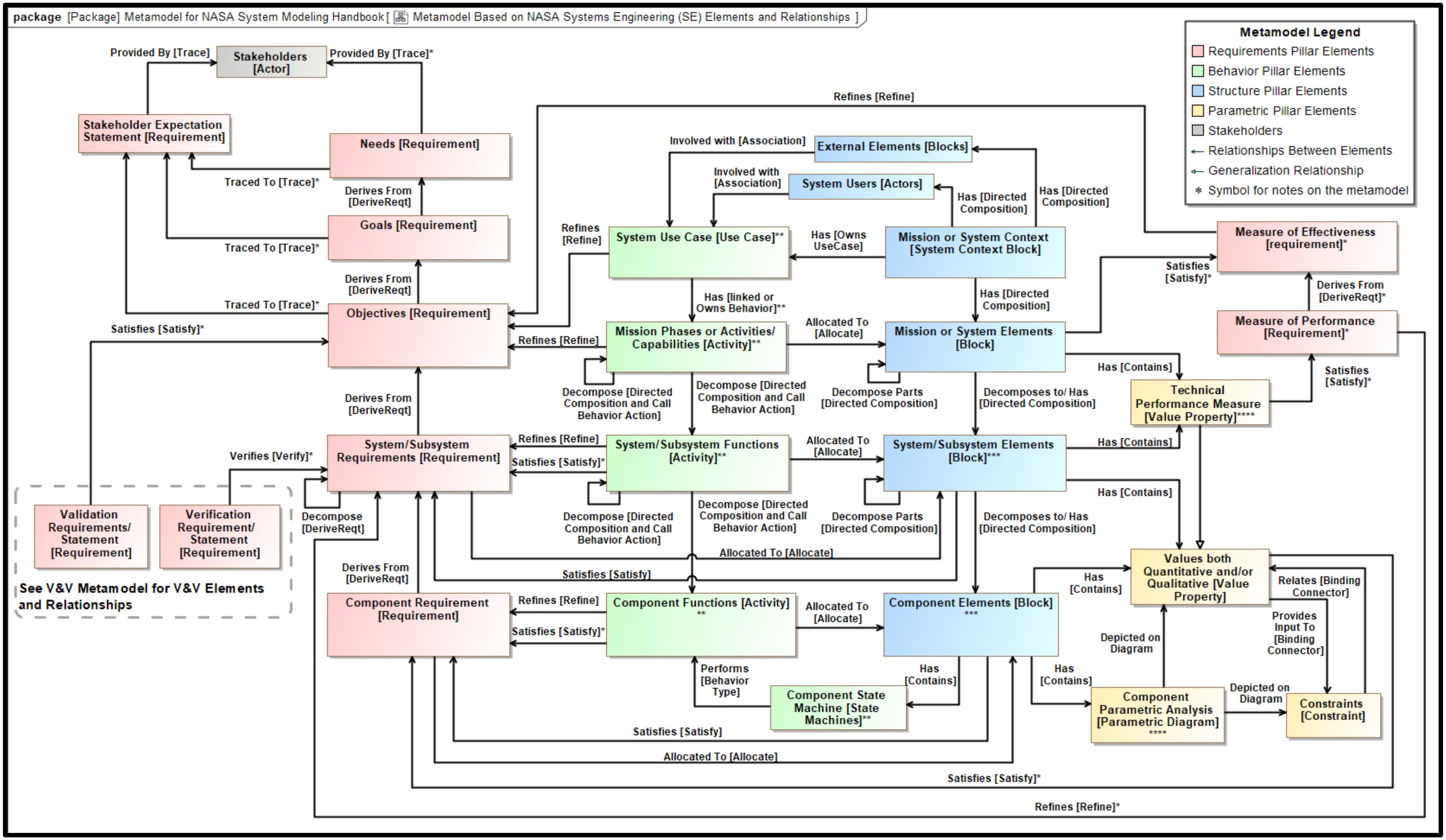


Figure 7-1 Metamodel Based on NASA Systems Engineering (SE) Elements and Relationships

Notes on the Metamodel:

***Notes on Requirements Pillar Elements:**

- In many cases requirements can be satisfied by a block; however, requirements can also be satisfied by behavior elements and value properties when the requirements are a performance requirement of a functional requirement (a value property can satisfy a performance requirement and a function can satisfy a functional requirement).
- Stakeholders can influence requirements at any level; hence the trace can exist at any level as shown with the Stakeholder Expectation Statement (In the metamodel, "Stakeholders" trace to "Needs" is shown for simplicity).
- The "satisfies" relationship for the MOE can point to different elements in the model depending on the content of the MOE (ex: behavior, structure, or parametric element, like a value property). The MOE can have multiple "satisfies" relationships.
- MOEs can occur at different levels of the design (for example: Subsystem elements).
- MOPs can occur at different levels of the design; they can derive from any requirement, MOE, and higher level MOP. MOPs can refine requirements similar to how the MOE refines the objectives.
- The "verifies" relationship from the Verification Requirements/Statement element traces to multiple requirement levels than depicted.
- The "satisfies" relationship from the Validation Requirements/Statement element can trace to different elements at different levels than just the objectives demonstrated here. Some examples include Stakeholder Expectation Statements and requirements at other levels.

****Notes on Behavior Pillar Elements:**

- Behaviors and interactions at all levels can use any of the SysML® Behavior Diagrams (uc, act, sd, and stm); these diagrams can be decomposed at each level to better articulate the expected behavior and interactions. For example, State Machines are applicable at each level (including at the Mission Level); they are shown at the component level for simplicity. Use Cases are shown at the Mission level for simplicity but can be applicable at each level.
- The association between the "Mission Use Case" and "Mission Phases and Activities" is OOSEM and MBSE Grid Supported. To support use case traceability, a stronger relationship can be used, for example, Dependency or Trace or Refine.

*****Notes on the Structural Pillar Elements:**

- From System to Component, decomposition happens in the same manner. Decompose to whatever level is needed for the project; do not go further than needed. Systems may decompose to additional Systems, Subsystems may decompose to additional Subsystems, and there may be an assembly level, etc.

******Notes on Parametric Pillar Elements:**

- Parametric Diagrams are applicable at other levels of decomposition not just at the component level.
- The TPM is one or more value properties that can occur at multiple levels to meet various MOPs defined; TPMs can be a parameter defined on a block or a parameter determined by computation.

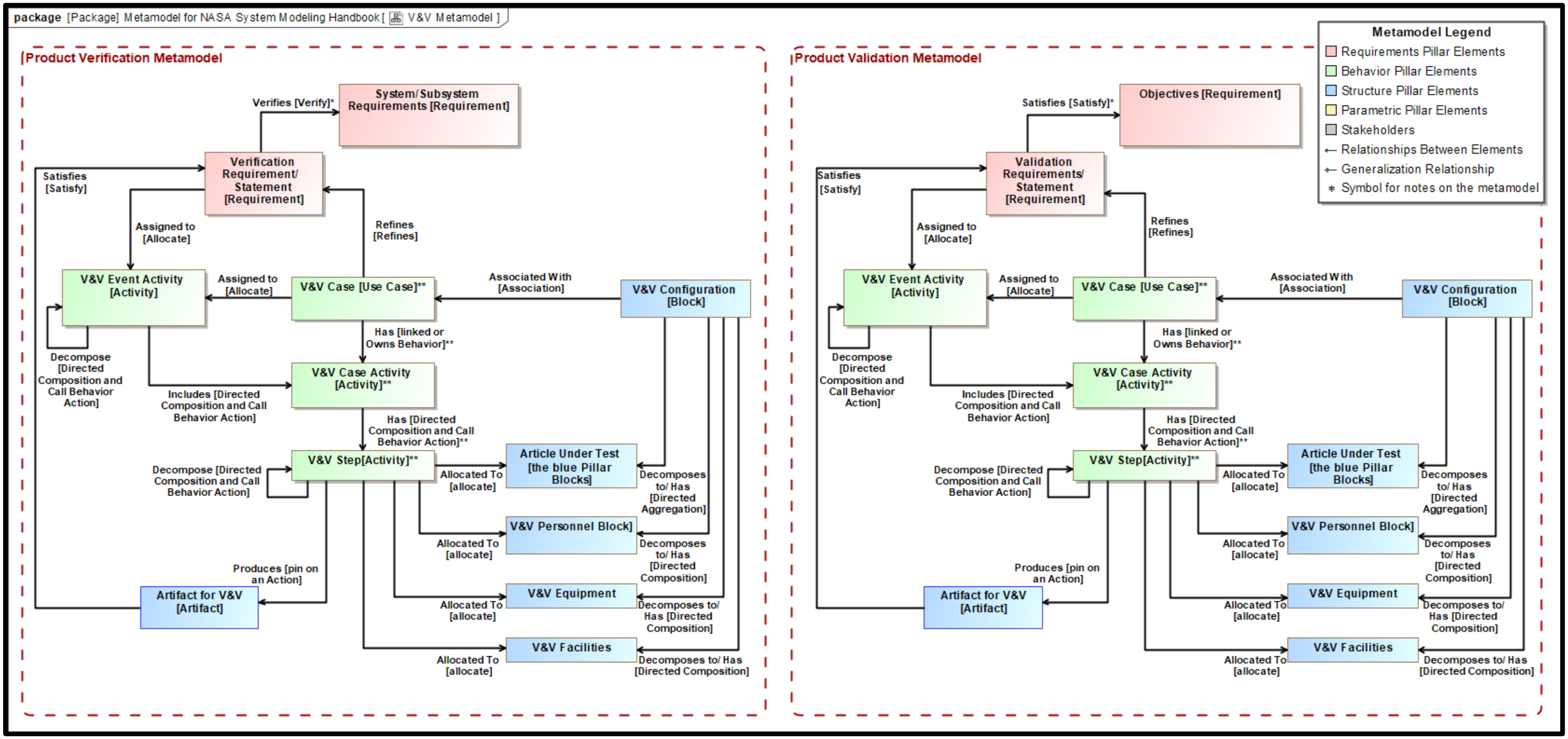


Figure 7-2 V&V Metamodel Based on NASA Systems Engineering (SE) Elements and Relationships

Notes on the V&V Metamodel:

- The metamodel shows V&V named elements for both Verification and Validation metamodels. Although Verification and Validation are distinct activities for each process, they follow a similar metamodel pattern. Note that V&V elements can be unique to just Verification or Validation or could be used to support both processes depending on the V&V case/activity. For example, a lot of Validation can be accomplished by Verification events/activities (in those cases a validation event or activity may leverage activities already captured in a verification description).

***Notes on Requirements Pillar Elements:**

- The “verifies” relationship from the Verification Requirements/Statement element traces to multiple requirement levels than depicted.
- The “satisfies” relationship from the Validation Requirements/Statement element can trace to different elements at different levels than just the objectives demonstrated here. Some examples include Stakeholder Expectation Statements and requirements at other levels.

****Notes on Behavior Pillar Elements:**

- Behaviors and interactions at all levels can use any of the SysML® Behavior Diagrams (uc, act, sd, and stm); these diagrams can be decomposed at each level to better articulate the expected behavior and interactions.
- The association between the "V&V Case" and "V&V Case Activity" can be depicted using a stronger relationship, for example, Dependency or Trace or Refine.

The metamodel in **Figure 7-1** and **Figure 7-2** is one approach to modeling in support of the NASA SE Engine. Within NASA, there are varying approaches to implement the metamodel. For example, the Property-Based Requirements (PBR) modeling approach can be applied to represent numerical requirements (see Appendix D.2). Another example of a variation to the metamodel is how relationships to subsystems are captured (example representations include reference properties or abstraction relationships). The intent is to have a method in this Handbook to support the objectives of generating SE products and enable tailoring of the metamodel to the program/project modeling methods as needed.

8. BUILDING THE MODEL

This section provides example SysML® diagrams and tables following the metamodel depicted in section 7, **Figure 7-1** and **Figure 7-2**. The diagrams and tables can be modeled in any order to support the SE activities on a program/project. SE activities can start at various points on the NASA SE Engine. For more information on the NASA SE Engine and NASA SE Processes, refer to NPR 7123.1 and NASA/SP-2016-6105. Section 9 of this Handbook provides details on diagrams and tables that can be used to support specific NASA SE Work Products for Stakeholder Expectations Definition, Requirements, and V&V.[10]

8.1 Stakeholders Identification and Expectations Statements bdd and Tables

An example Stakeholders Identification and Expectations Statements Block Definition Diagram (bdd) along with the corresponding metamodel is shown in **Figure 8.1-1**, **Stakeholder Expectations Metamodel (Left); bdd (Right)**. The stakeholders are captured as actor elements and the stakeholder expectation statements are depicted with a trace relationship to the stakeholder.

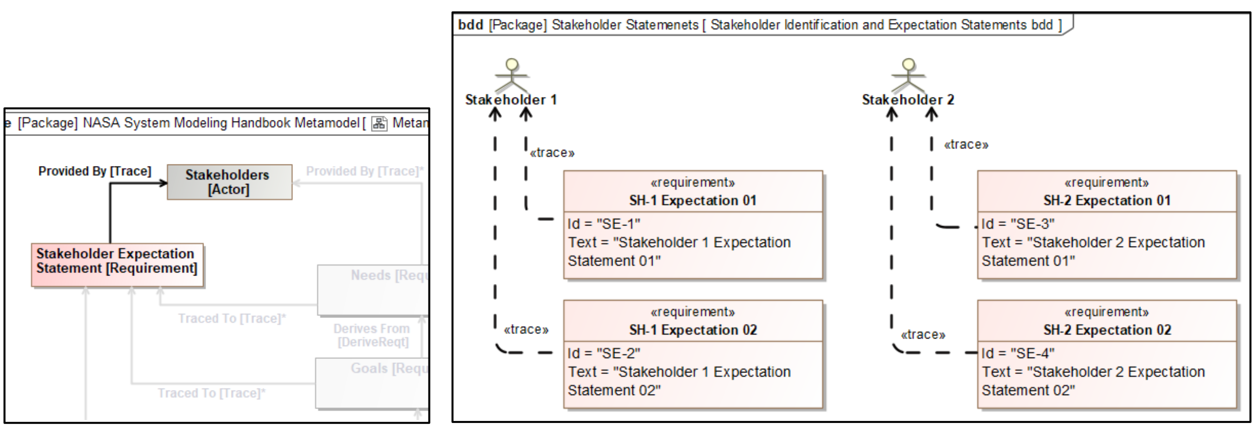


Figure 8.1-1 Stakeholder Expectations Metamodel (Left); bdd (Right)

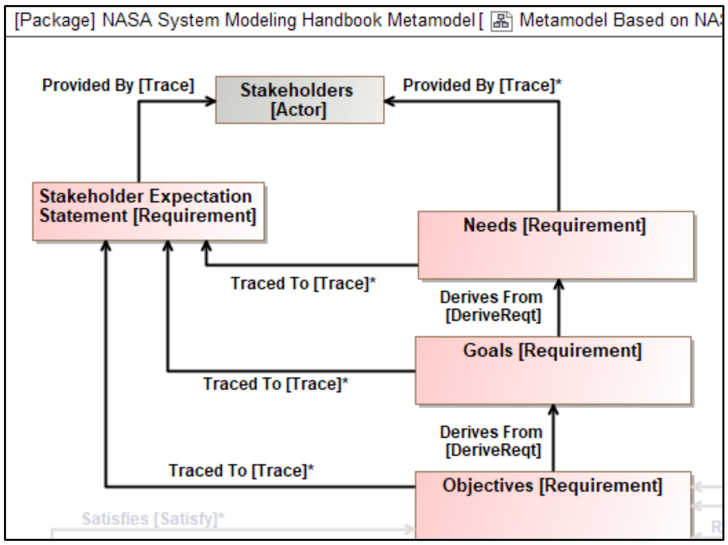
Figure 8.1-2, **Stakeholder Description Table**, shows an example table that lists the stakeholders along with a description.

[10] Modeling tool used for Diagrams and Tables is CATIA® No Magic (a Dassault Systemes Product)

#	Name	Documentation
1	Stakeholder 1	Stakeholder Description....
2	Stakeholder 2	Stakeholder Description....

Figure 8.1-2 Stakeholder Description Table

Another example table that can support Stakeholder identification and Expectation Statements is shown in **Figure 8.1-3**, **Stakeholder Expectation Statement and Traceability Metamodel (Top); Table (Bottom)**. This table depicts Needs, Goals, and Objectives that trace to the Stakeholder Expectation Statement.



#	Traced To	Name	Text	Traced From
1	Stakeholder 1	SH-1 Expectation 01	Stakeholder 1 Expectation Statement 01	need-1 Need
2	Stakeholder 1	SH-1 Expectation 02	Stakeholder 1 Expectation Statement 02	goal-1 Goal 1.1
3	Stakeholder 2	SH-2 Expectation 01	Stakeholder 2 Expectation Statement 01	goal-1 Goal 1.1
4	Stakeholder 2	SH-2 Expectation 02	Stakeholder 2 Expectation Statement 02	obj-1 Objective 1

Figure 8.1-3 Stakeholder Expectation Statement and Traceability Metamodel (Top); Table (Bottom)

Note: In the metamodel, "Stakeholders" trace to "Needs" is shown for simplicity; however, stakeholders can influence requirements at any level similar to how the trace is shown between Stakeholder Expectation Statements and the Needs, Goals, and Objectives.

The table shows the Needs, Goals, and Objectives as traced to the Stakeholder Expectation Statements. A similar table can be created for stakeholders and the direct trace to the Needs, Goals, and Objectives.

8.2 Requirements Diagram (req) of Needs, Goals, and Objectives (NGOs)

An example requirements diagram of NGOs is shown in **Figure 8.2-1**, **NGO Metamodel (Left); req of NGOs (Right)**. The metamodel portion of the NGOs from section 7, **Figure 7-1**, is shown on the left. A sample SysML® requirements diagram of the Needs, derived Goals, and derived Objectives is shown on the right.

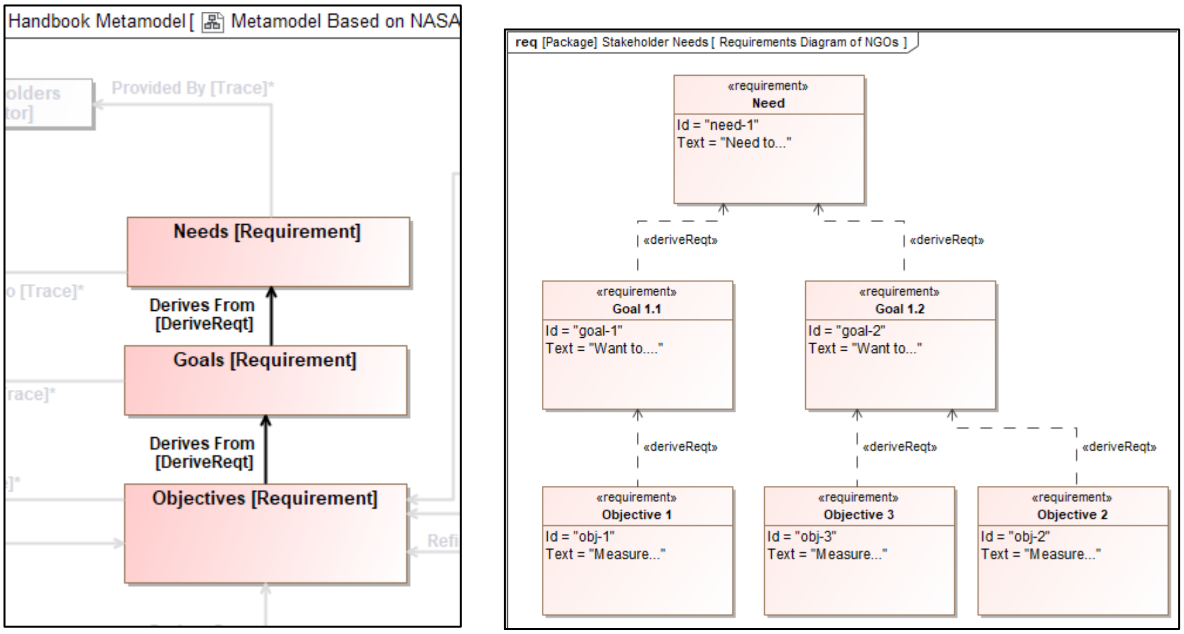
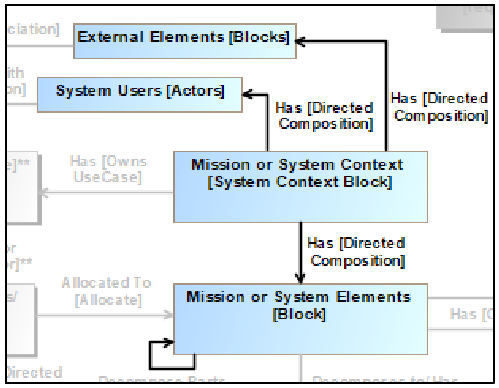


Figure 8.2-1 NGO Metamodel (Left); req of NGOs (Right)

8.3 System Context Block Definition Diagram (bdd)

An example System Context bdd is shown in **Figure 8.3-1**, **System Context Metamodel (Top); bdd (Bottom)**. The system context depicts the scope and boundaries of the system being modeled and includes the system of interest, the system users, and the external system elements that interface with the system of interest. The system context can be captured using block definition diagrams (bdd) and internal block diagrams (ibd) (see section 8.4 for the ibd representation). The metamodel portion of the system context from section 7, **Figure 7-1**, is shown in the top of **Figure 8.3-1**. A sample system context bdd of *System XYZ* as the system of interest is shown on the bottom.

Note: System Context can depict any level of structure to support the project/program. For example, the System Context can depict a mission, instrument level, subsystem level, or payload level.



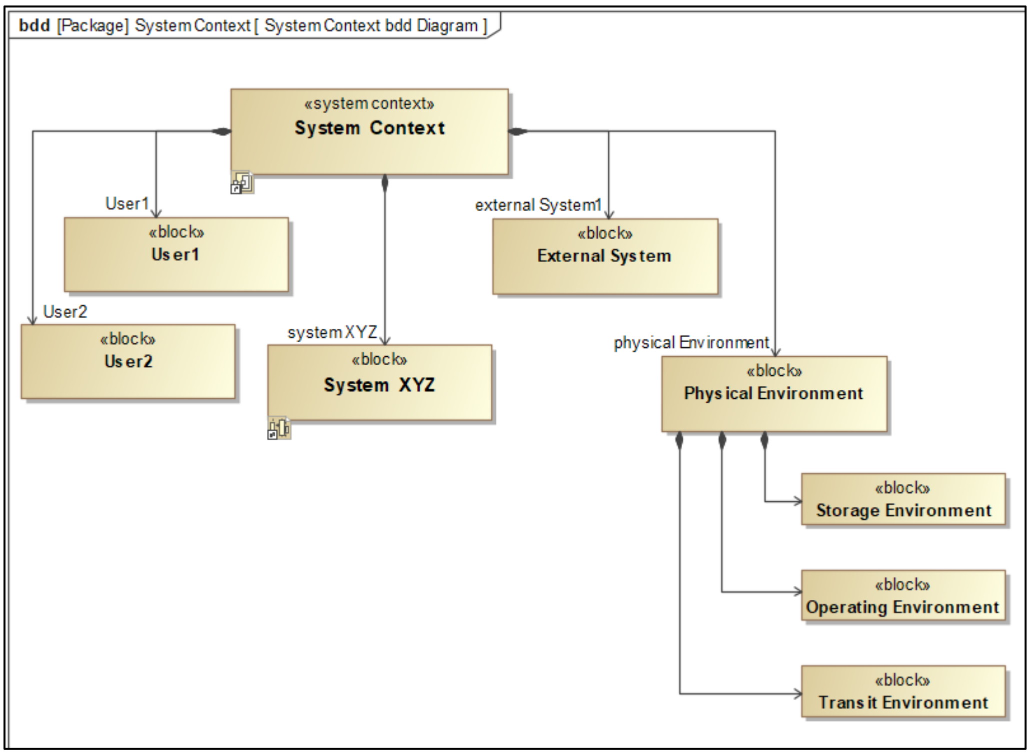


Figure 8.3-1 System Context Metamodel (Top); bdd (Bottom)

8.4 System Context Internal Block Diagram (ibd)

An example System Context ibd is shown in **Figure 8.4-1**, **System Context Metamodel (Top); ibd (Bottom)**. The system context can be captured using block definition diagrams (bdd) and internal block diagrams (ibd) (see section 8.3 for the bdd representation). An ibd can be used to show how structure elements interface. The metamodel portion of the system context from section 7, **Figure 7-1**, is shown in the top of **Figure 8.4-1**. A sample ibd of the *System Context* block in **Figure 8.3-1** is shown in **Figure 8.4-1** on the bottom. This sample ibd shows the interfaces between System XYZ, User1, external System1, and the physical environment and items that flow across those interfaces. See Appendix E for interface metamodel details.

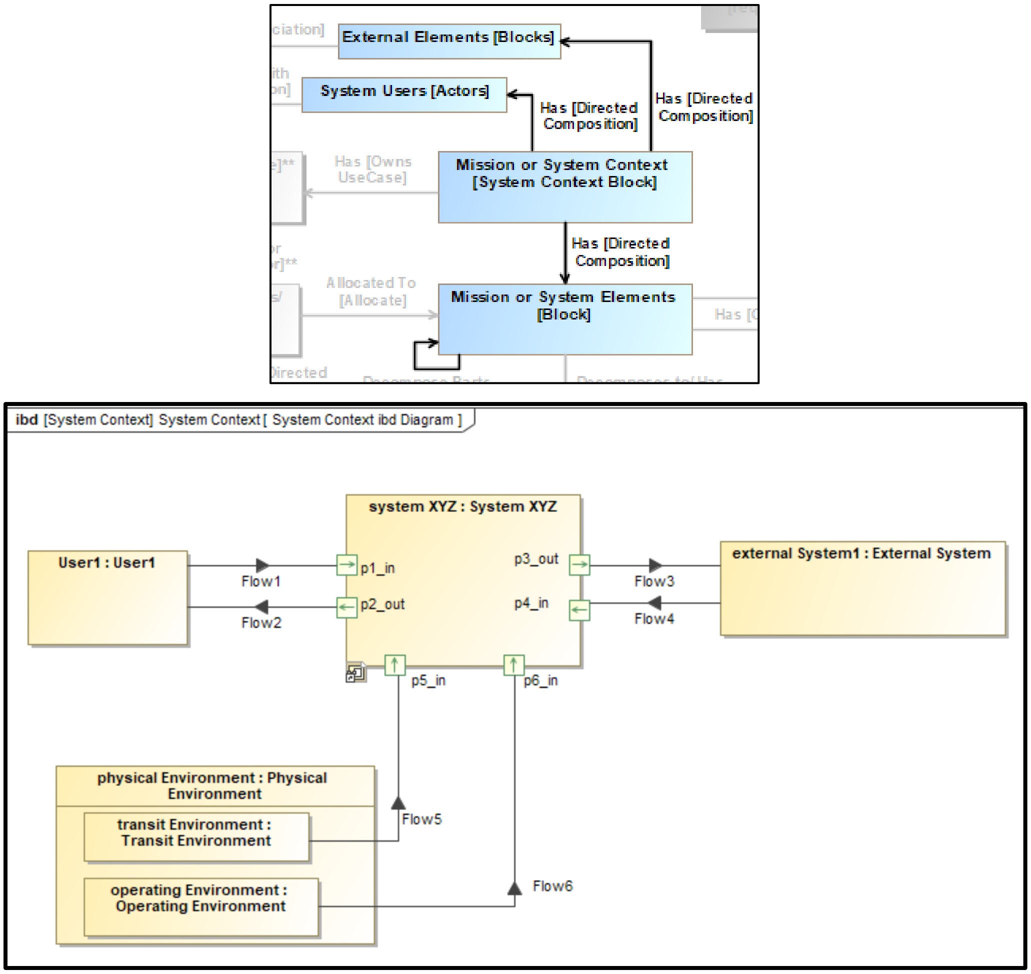


Figure 8.4-1 System Context Metamodel (Top); ibd (Bottom)

Note:

- The System Context ibd depicts different examples of how to illustrate information flow using ports or without ports. The modeler can display port types or omit them for visual simplicity depending on how the diagram is being used.
- See Appendix E for metamodel details for interface modeling.

8.5 System Use Case (uc) Diagram

An example use case diagram is shown in **Figure 8.5-1**, **System Use Case Metamodel (Left); uc (Right)**. Use case diagrams describe the functions of a system and the interactions between those functions and System actors or elements. The metamodel portion of the System use case from Section 7, **Figure 7-1**, is shown on the left. A sample SysML® use case diagram is shown on the right.

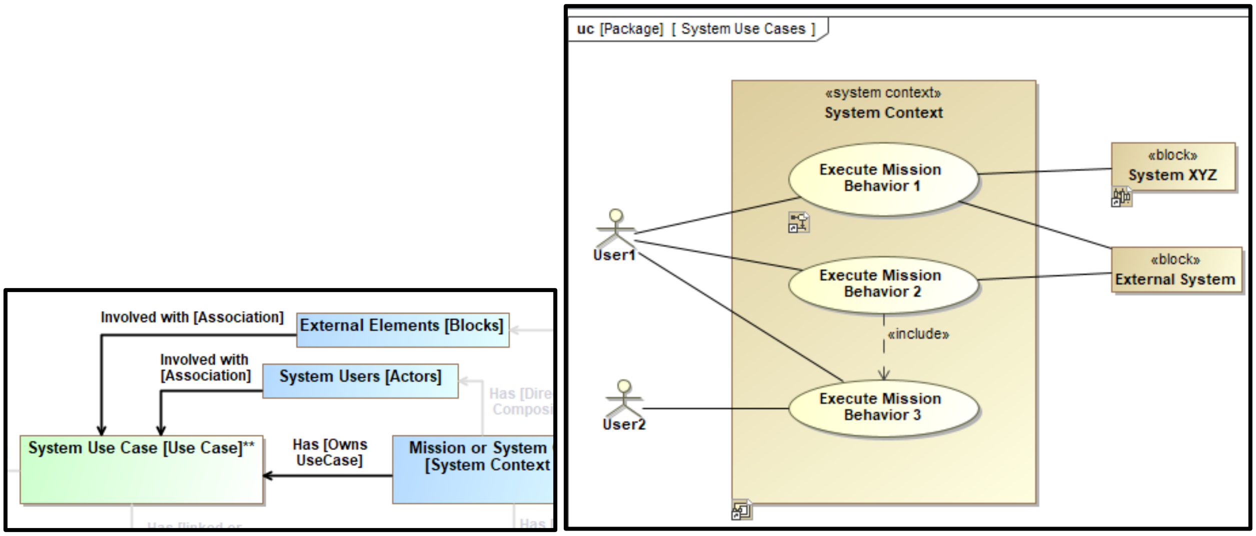


Figure 8.5-1 System Use Case Metamodel (Left); uc (Right)

Notes:

- Actors can represent roles played by human users, external hardware, or any other subjects.
- The <<include>> relation is used to indicate where one Use Case may be a subset of another.
- An example of the actor "User1" depicted in this Use Case diagram could be an automation system.
- An example of the Use Case "Execute Mission Behavior 1" could be "Fly a desired trajectory."
- An example of "System XYZ" could be the spacecraft or aircraft of interest.

8.6 Activity Diagram (act) Supporting Use Case

An example activity diagram is shown in **Figure 8.6-1**, **Activity Elements Allocated to Structure Elements Metamodel (Top); act of Execute Mission Behavior 1 Use Case (Bottom)**. The activity diagram (act) is one of the four behavior diagrams used to describe a system's behavior. In this example, the activity diagram is used to further explain the details of the use case example "Execute Mission Behavior 1" (see section 8.4 on use case diagrams). Activities can be captured in activity diagrams showing interactions between activities and allocations to structure elements (see Appendix E). The metamodel portion of activity elements and their relationships from section 7, **Figure 7-1**, is shown in the top of **Figure 8.6-1**. A sample SysML® activity diagram using swim lanes to allocate activity elements *User Activity 1 and 2*, *System Function 1* and *System Function 2,* and *External System Activity 1* to structure elements *User1, System XYZ*, and *External System* is shown on the bottom. Note: Use of the ':' in the action name of the activity diagram indicates the activity typing of the action; Without the colon, the action only exists within this diagram.

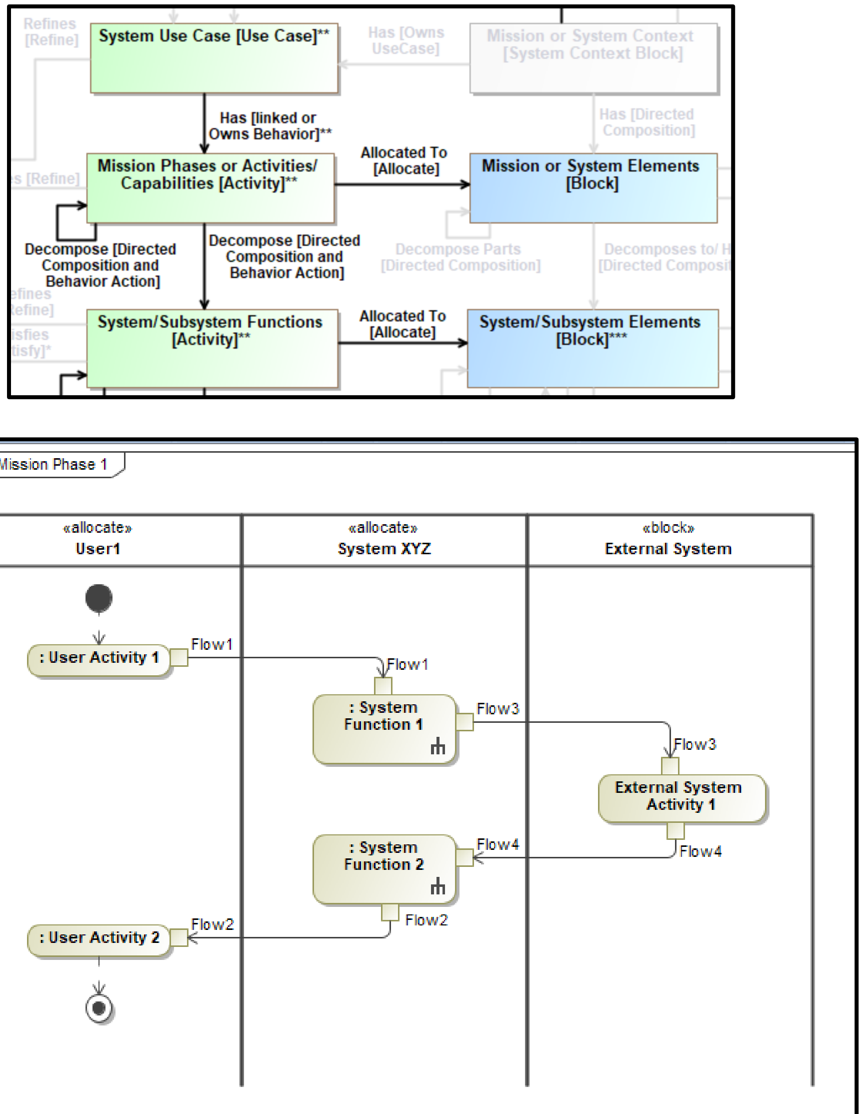


Figure 8.6-1 Activity Elements Allocated to Structure Elements Metamodel (Top); act of Execute Mission Behavior 1 Use Case (Bottom)

8.7 Structural Decomposition Block Definition Diagram (bdd)

An example structural decomposition bdd is shown in **Figure 8.7-1**, **Structural Decomposition Metamodel (Left); bdd (Right)**. The metamodel portion of the structure decomposition from section 7, **Figure 7-1**, is shown on the left. A sample SysML® bdd of a system decomposition is shown on the right.

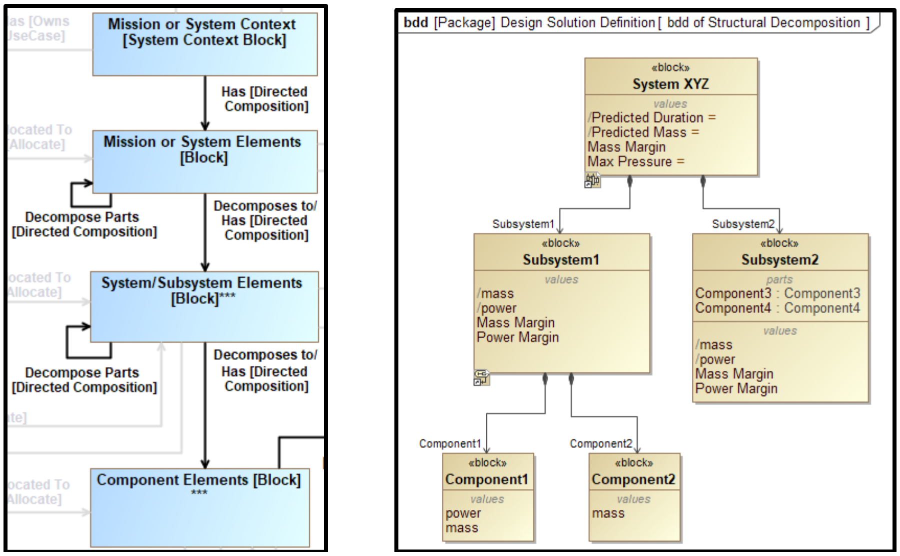


Figure 8.7-1 Structural Decomposition Metamodel (Left); bdd (Right)

8.8 Internal Block Diagram (ibd) of Structure Interconnections

An example ibd of structure interconnections is shown in **Figure 8.8-1**, **Structural Decomposition Metamodel (Left); ibd (Right)**. The interfaces between structure elements can be captured in an internal block diagram (ibd). The metamodel portion of the structure decomposition from section 7, **Figure 7-1**, is shown on the left. A sample ibd of the *System XYZ* block from **Figure 8.7-1** and the interfaces between *Subsystem 1* and *Subsystem 2* are shown on the right. See Appendix E for Interface metamodel details.

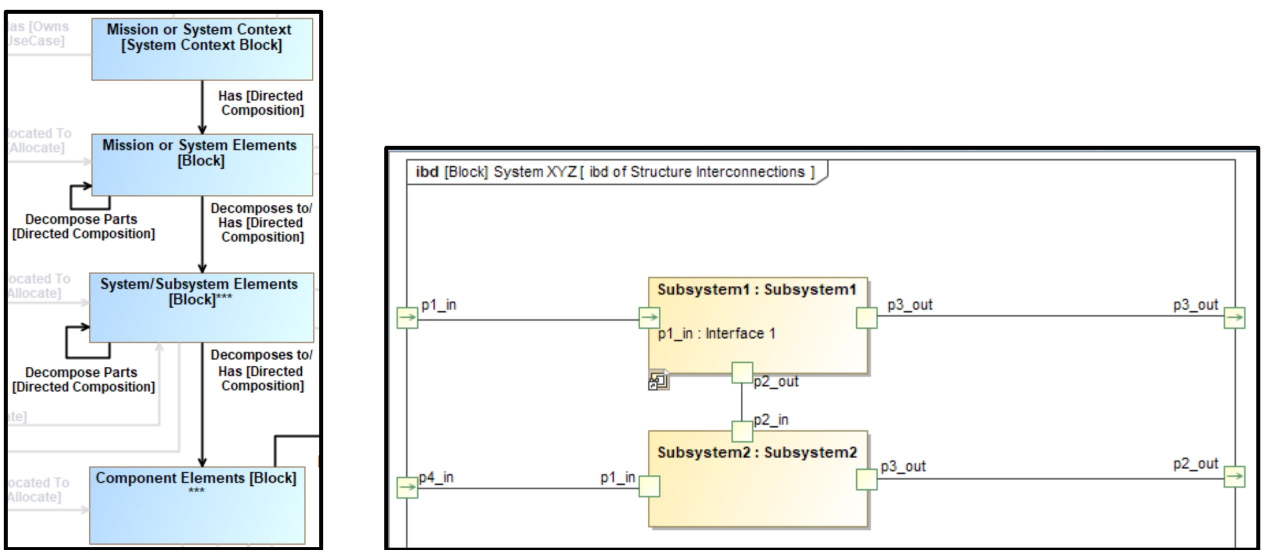


Figure 8.8-1 Structural Decomposition Metamodel (Left); ibd (Right)

Notes: The diagram shows a few different port depictions. The modeler can display port types or omit them for visual simplicity depending on how the diagram is being used. Details about the port types can be captured in supporting tables as well (see ConOps Template in Appendix F, section 3.3 Interfaces for example tables.

8.9 Functional Decomposition of Activities via a Block Definition Diagram (bdd)

An example functional decomposition of activities via a bdd is shown in **Figure 8.9-1**, **Functional Decomposition Metamodel (Left); bdd (Right)**. The same activities highlighted in the activity diagram in section 8.6 can be represented in a bdd to depict functional decomposition. **Figure 8.9-1** shows the metamodel portion of the behavior decomposition from section 7, **Figure 7-1**, on the left and a sample SysML® bdd of functional decomposition of activities on the right.

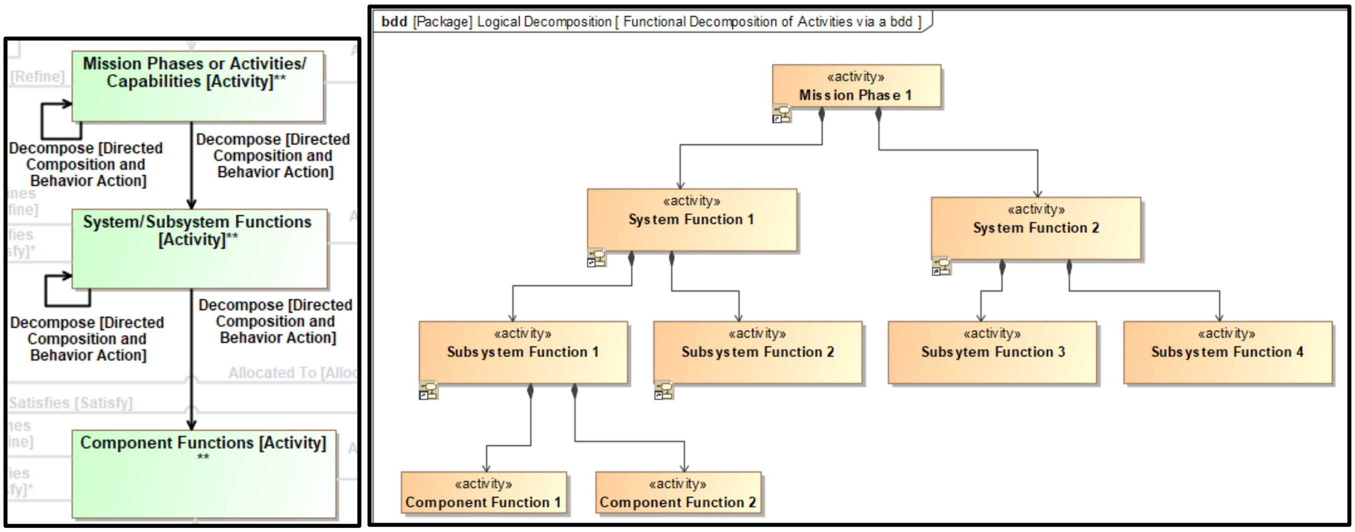


Figure 8.9-1 Functional Decomposition Metamodel (Left); bdd (Right)

8.10 System Requirement Diagram (req)

An example system requirement diagram is shown in **Figure 8.10-1**, **Requirements Metamodel (Left); System req (Right)**. The metamodel portion of requirements and their relationships from section 7, **Figure 7-1**, is shown on the left. A sample SysML® requirements diagram of the system requirements decomposition and flow-down using derived requirement relationship is shown on the right.

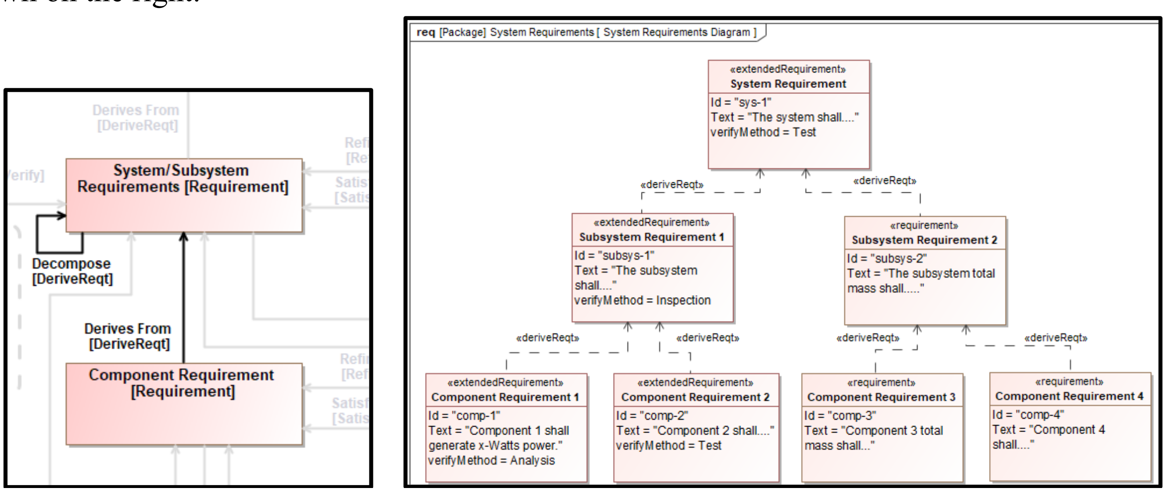


Figure 8.10-1 Requirements Metamodel (Left); System req (Right)

8.11 Requirements Table with Traceability

A Requirements Table is a tabular format used to represent requirements, their properties, and relationships.[11] The metamodel portion of requirements and their relationships from section 7, **Figure 7-1**, is shown in **Figure 8.11-1**, **Requirements and Traceability Metamodel**. The tabular view of the requirements, their properties, and relationships are shown in **Figure 8.11-2**, **Example Requirements Table with Traceability**.

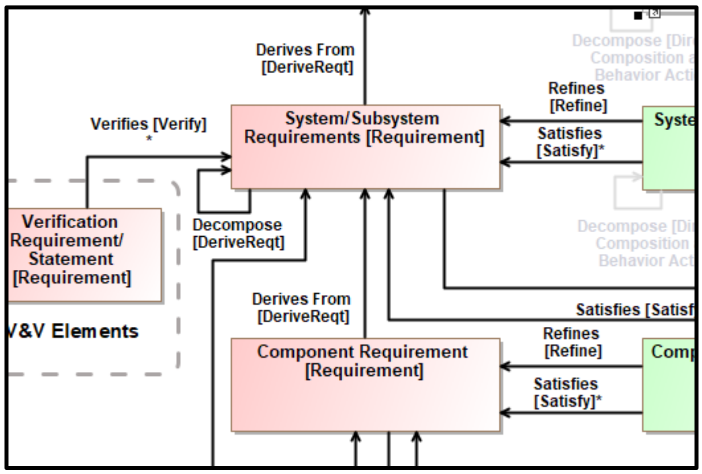


Figure 8.11-1 Requirements and Traceability Metamodel

#	Id	Name	Text	Derived From	Refined By	Allocated To	Verify Method	Verified By	Satisfied By
1	sys-1	System Requirement	The system shall....		System Function 1(context System X	System XYZ	Test	ver-1 Verification Requirement 1	System XYZ
2	subsys-1	Subsystem Requirement 1	The subsystem shall....	sys-1 System Requirement	Subsystem Function 1(context Subsy Subsystem Function 2(context Subsy		Inspection	ver-2 Verification Requirement 2	Subsystem1
3	subsys-2	Subsystem Requirement 2	The subsystem total mass shall.....	sys-1 System Requirement	Subsytem Function 3(context Subsy Subsystem Function 4(context Subsy MOP-2.2 (MOP) Subsystem 2 Mass				/mass
4	comp-1	Component Requirement 1	Component 1 shall generate x-Watts power.	subsys-1 Subsystem Requirement 1	Component Function 1		Analysis	ver-3 Verification Requirement 3	power value
5	comp-2	Component Requirement 2	Component 2 shall....	subsys-1 Subsystem Requirement 1	Component Function 2		Test		Component2

Figure 8.11-2 Example Requirements Table with Traceability

8.12 Requirements Diagram (req) of MOEs, MOPs, and Requirements Traceability

An example traceability between Measures of Effectiveness (MOEs), Measures of Performance (MOPs) and system requirements and objectives is shown in **Figure 8.12-1**, **MOE and MOP Traceability Metamodel (Top); req of MOEs, MOPs, and Requirements Traceability (Bottom)**. The metamodel portion of MOEs and MOPs from section 7, **Figure 7-1**, is shown on the top. A sample SysML® requirements diagram of MOEs and related MOPs along with traceability to objectives and system requirements is shown on the bottom. The information from the diagram can be summarized in a table as seen in **Figure 8.12-2**, **Requirements Table of Objectives and Trace to MOEs and MOPs**.

[11] Object Management Group (OMG). (2024). "System Modeling Language (SysML), Version 1.7." (https://sysml.org/sysml-specs/)

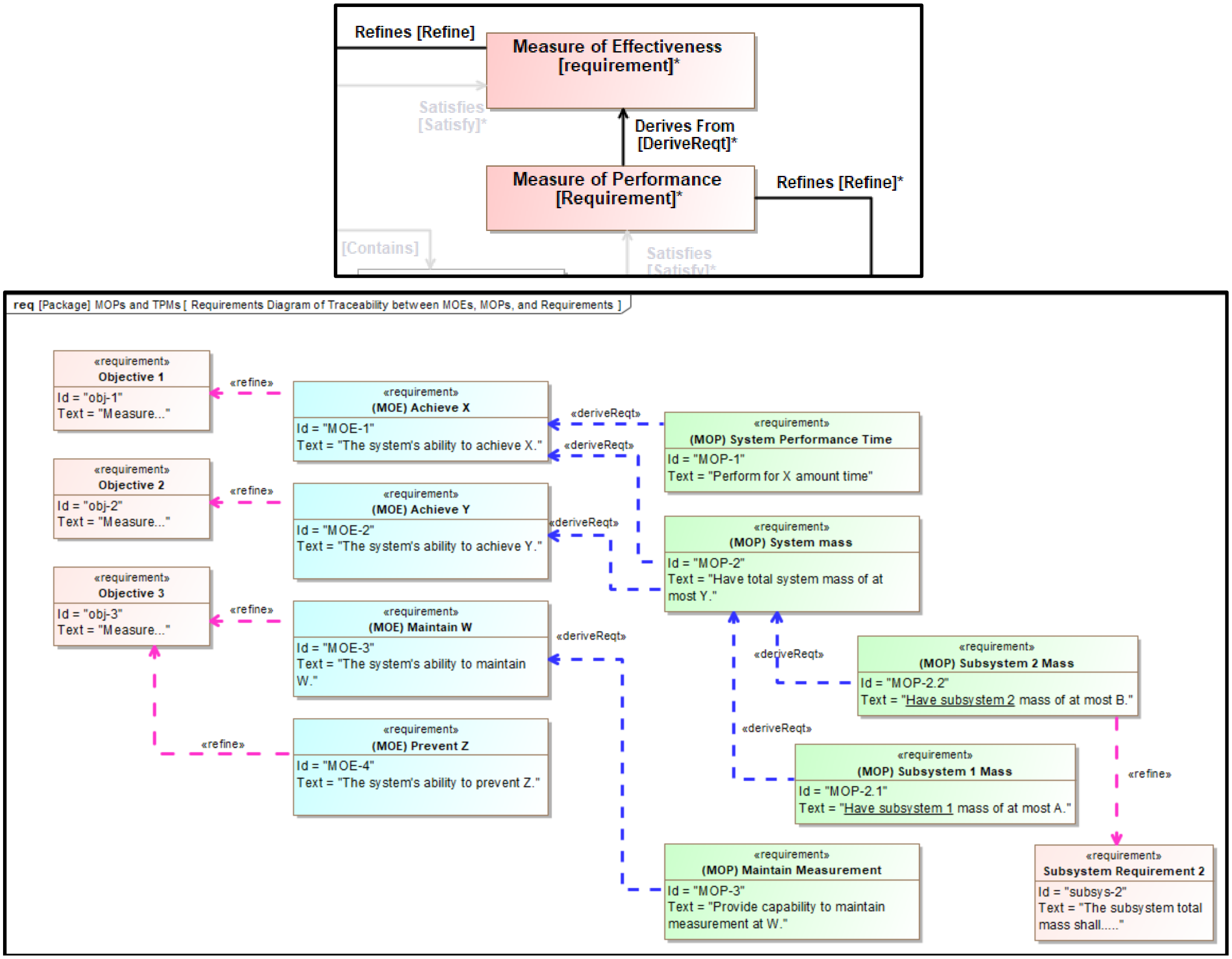


Figure 8.12-1 MOE and MOP Traceability Metamodel (Top); req of MOEs, MOPs, and Requirements Traceability (Bottom)

Notes:
- Different projects may derive their requirements from MOPs, Objectives, or Level 1/Level 2/Level 3 requirement breakdowns. This example shows how the model constructs can be captured to depict relationships between the various elements.
- MOE to MOP is not a one-to-one relationship.
- MOEs can occur at different levels of the design (for example: Subsystem elements).
- MOPs can occur at different levels of the design; they can derive from any requirement, MOE, and higher level MOP.
- MOPs can refine requirements similar to how the MOE refines the objectives.

Figure 8.12-2 provides an example tabular representation of the MOEs and MOPs related to objectives 1 thru 3 in the requirement diagram.

#	Name	Text	△ Refined By	Traced MOPs
1	obj-1 Objective 1	Measure...	MOE-1 (MOE) Achieve X	MOP-1 (MOP) System Performance Time MOP-2 (MOP) System mass
2	obj-2 Objective 2	Measure...	MOE-2 (MOE) Achieve Y	MOP-2 (MOP) System mass
3	obj-3 Objective 3	Measure...	MOE-4 (MOE) Prevent Z MOE-3 (MOE) Maintain W	MOP-3 (MOP) Maintain Measurement

Figure 8.12-2 Requirements Table of Objectives and Trace to MOEs and MOPs

8.13 Structural Decomposition bdd with MOP and TPM Identification

An example bdd showing Technical Performance Measure (TPM) values and traceability to Measures of Performance (MOPs) is shown in **Figure 8.13-2**, **bdd with MOPs and TPM Identification**. The metamodel portion of structural decomposition, TPMs and MOPs from section 7, **Figure 7-1**, is shown in **Figure 8.13-1, Structural Decomposition, MOP, and TPM Metamodel bdd.** The information from the bdd can be summarized in a table as seen in **Figure 8.13-3**, **Table of MOPs with TPMs and Traceability**.

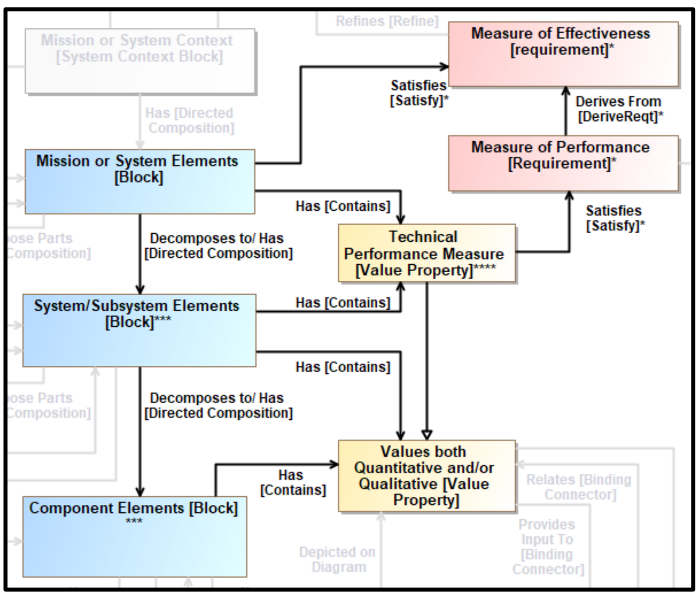


Figure 8.13-1 Structural Decomposition, MOP, and TPM Metamodel bdd

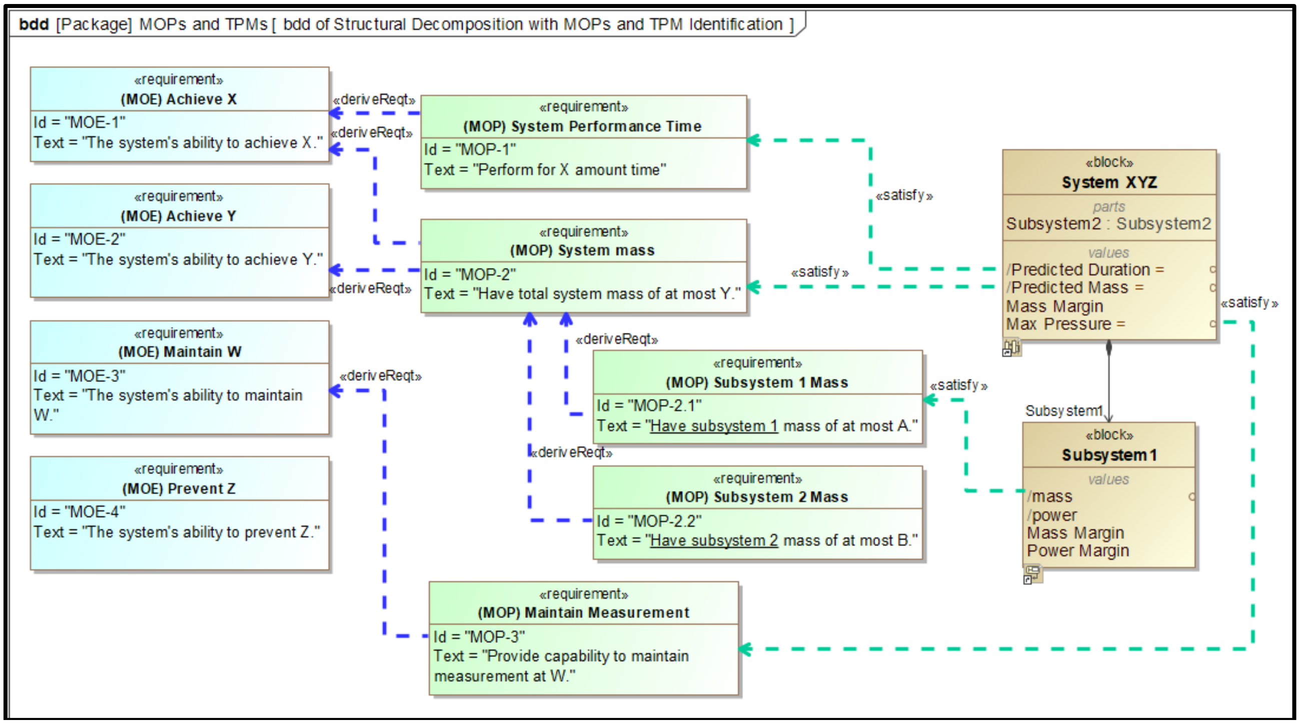


Figure 8.13-2 bdd with MOPs and TPM Identification

Notes:

- MOE to MOP is not necessarily a one-to-one relationship.
- MOPs can occur at different levels of the design; they can derive from any requirement, MOE, and higher level MOP.
- The TPM is one or more value properties that can occur at multiple levels to meet various MOPs defined.
- TPMs can be a parameter defined on a block or a parameter determined by computation (values can be marked as derived via a "/", as shown for the Predicted Duration and Predicted Mass values in System XYZ).

Figure 8.13-3 provides an example tabular representation of the MOPs, traced TPMs and the structural element owning the TPM.

#	△ Id	Name	Text	Derived From	TPMs (via Satisfied By)	Element that Owns TPMs
1	MOP-1	(MOP) System Performance Time	Perform for X amount time	MOE-1 (MOE) Achieve X	/Predicted Duration =	System XYZ
2	MOP-2	(MOP) System mass	Have total system mass of at most Y.	MOE-2 (MOE) Achieve Y MOE-1 (MOE) Achieve X	/Predicted Mass =	System XYZ
3	MOP-2.1	(MOP) Subsystem 1 Mass	Have subsystem 1 mass of at most A.	MOP-2 (MOP) System mass	/mass	Subsystem1
4	MOP-2.2	(MOP) Subsystem 2 Mass	Have subsystem 2 mass of at most B.	MOP-2 (MOP) System mass	/mass	Subsystem2
5	MOP-3	(MOP) Maintain Measurement	Provide capability to maintain measurement at W.	MOE-3 (MOE) Maintain W	Max Pressure =	System XYZ

Figure 8.13-3 Table of MOPs with TPMs and Traceability

8.14 Requirements Diagram (req) of Verification Requirements/ Statements

An example requirements diagram of verification requirements/ statements is shown in **Figure 8.14-1**, **Verification Requirements Metamodel (Left); Verification req (Right)**. The metamodel portion of the verification requirement/statement and the "verifies" relationship to a system requirement from section 7, **Figure 7-1**, is shown on the left. A sample requirements diagram (req) with verification requirements and verification attributes is shown on the right. In **Figure 8.14-1**, *System Requirement*, *Subsystem Requirement 1*, and *Component Requirement 1* have a verification attribute, *verifyMethod*. This property can also be seen in the example *Verification Requirement 1*, *2*, and *3.* The verify method attribute can be modeled on the requirements or the verification requirement. The modeler may choose to specify the verification method on the requirement in early life cycle phases and then to move to the verification requirement at later phases. Note the same information can also be depicted in a tabular format (see sections 8.16 and 8.17 for example tables).

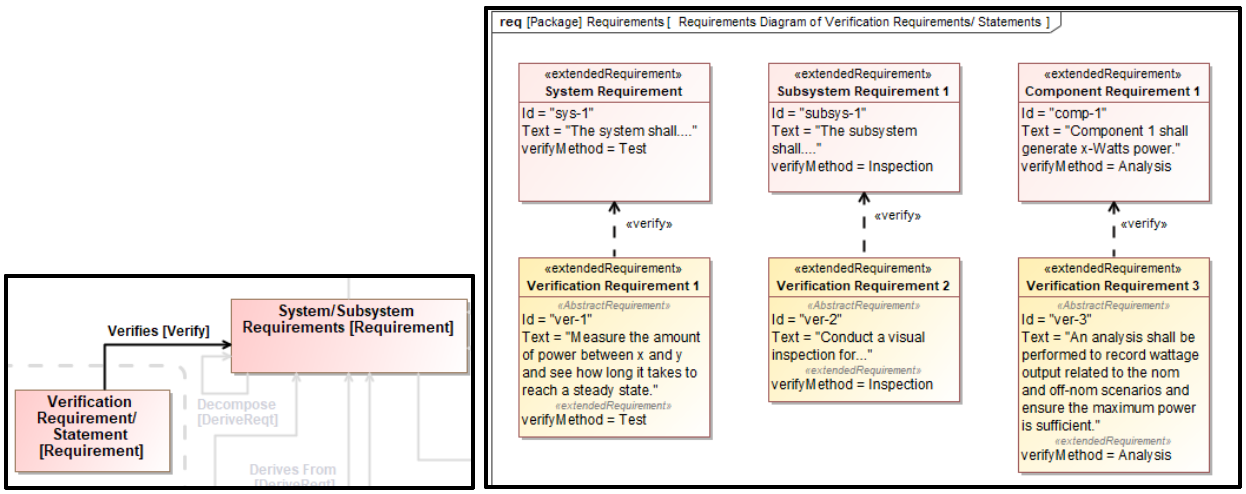


Figure 8.14-1 Verification Requirements Metamodel (Left); Verification req (Right)

Notes on the verify method attribute:

- The verify method attribute is a property of the extended requirement—a SysML® extension to requirements.
- The verify method attribute can be modeled on the requirements or the verification requirement. The handbook is showing the method called out in both places, however only one is needed.
- The modeler may choose to specify the verification method on the requirement in early life cycle phases and then to move to the verification requirement at later phases.
- There can be multiple verification requirements traced to a system requirement.

8.15 Requirements Diagram (req) of Verification Requirements and Traceability

Figure 8.15-1, Verification Requirements and Traceability Metamodel (Left); req (Right), provides an example verification requirements diagram that follows the V&V metamodel from section 7, **Figure 7-2**. The Verification metamodel is shown on left and an example requirements diagram depicting the related elements is provided on right. Note the same information can also be depicted in a tabular format (see sections 8.16 and 8.17 for example tables).

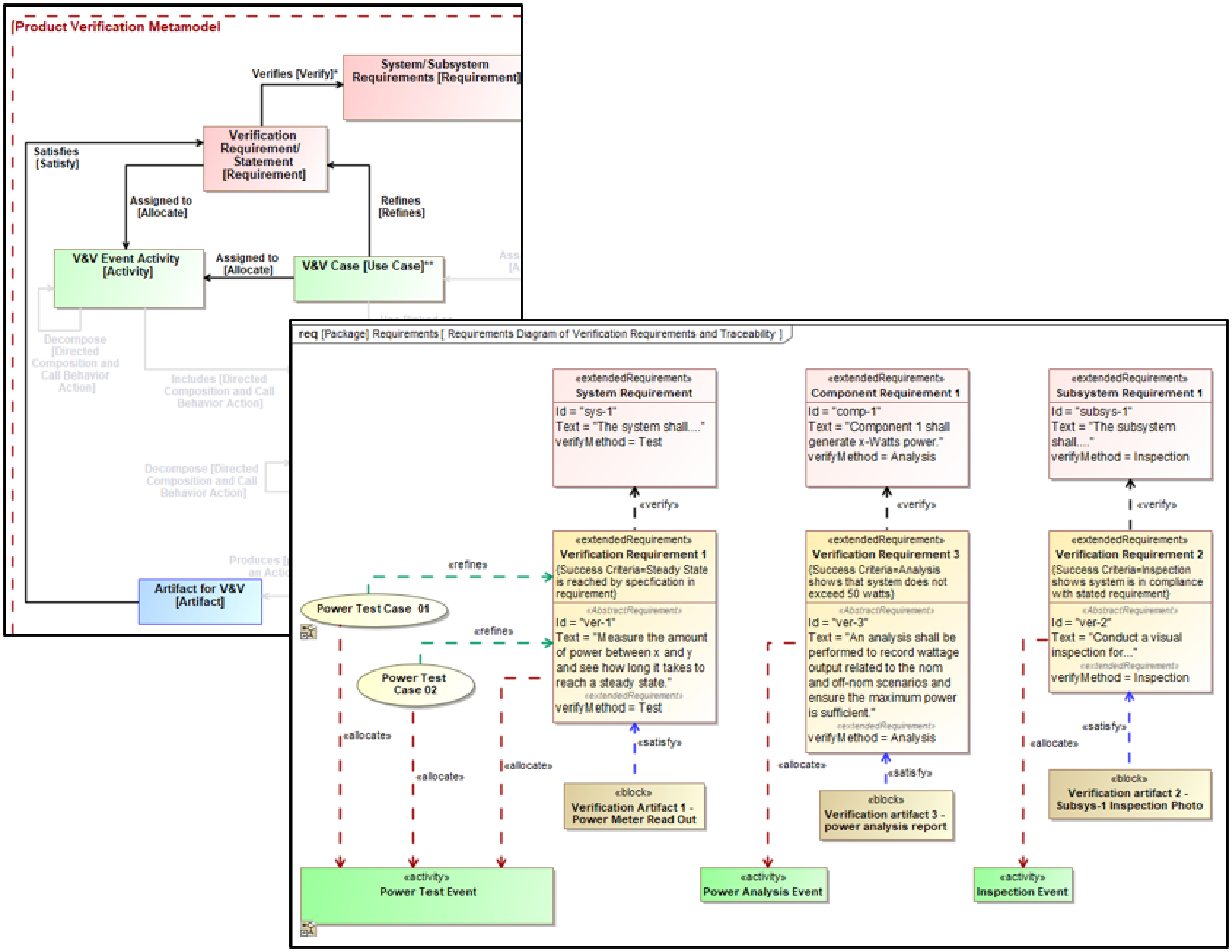
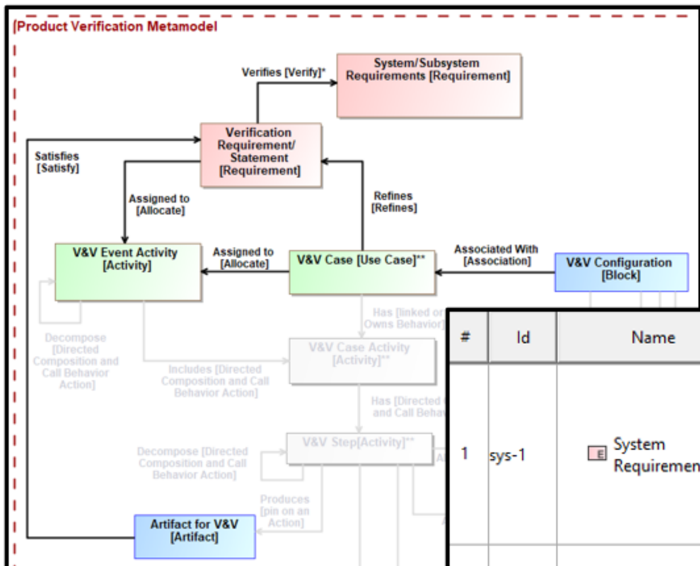


Figure 8.15-1 Verification Requirements and Traceability Metamodel (Left); req (Right)

> Notes:
> - The success criteria is captured as a constraint relationship on the verification requirements in this example, shown in { } below the *Verification Requirement 1, Verification Requirement 2,* and *Verification Requirement 3* elements.
> - Only *Verification Requirement 1* is showing V&V Cases in this example (as *Power Test Case 01* and *Power Test Case 02*); however, an analysis case and inspection case could be identified for the other 2 verification requirement examples.
> - Section 8.16 provides a Requirements Verification Matrix for this example.
> - Section 8.17 provides a Verification Requirements Matrix/ Verification Compliance Spreadsheet for this example.

8.16 Table for Requirements Verification Matrix

Figure 8.16-1, **Requirements Verification Matrix (Right); Metamodel (Left)**, provides an example tabular view of the content in the verification requirements diagram in **Figure 8.15-1**. Note: In this example, a table of system requirements is depicted with information on the verification requirements that relates to it by the “verifies” relationship and attributes and relationships of the verification requirement. Section 8.17 provides a table of the verification requirements and its relationships and attributes directly. For more information on the on Requirements Verification Matrix, refer to NASA/SP-2016-6105. The NASA SE Handbook provides a sample Requirements Verification Matrix as seen in **Figure 8.16-2**, **Sample Requirements Verification Matrix from the NASA SE Handbook**.



#	Id	Name	Text	Verified By	Verification Success Criteria	Verify Method	Verification Configuration	Artifact for Verification	Documentation/ Results
1	sys-1	System Requirement	The system shall....	ver-1 Verification Requirement 1	Steady State is reached by specfication in requirement	Test	Verification Configuration 1 - Power Test	Verification Artifact 1 - Power Meter Read Out	Closure Status: Closure Date: Closure Notes: Description of how the artifacts met the success criteria. And non-conformance findings and statuses
2	subsys-1	Subsystem Requirement 1	The subsystem shall....	ver-2 Verification Requirement 2	Inspection shows system is in compliance with stated requirement	Inspection		Verification artifact 2 - Subsys-1 Inspection Photo	
3	comp-1	Component Requirement 1	Component 1 shall generate x-Watts power.	ver-3 Verification Requirement 3	Analysis shows that system does not exceed 50 watts	Analysis		Verification artifact 3 - power analysis report	
4	comp-2	Component Requirement 2	Component 2 shall....			Test			

Figure 8.16-1 Requirements Verification Matrix (Right); Metamodel (Left)

Comparison to NASA SE Handbook Example Requirements Verification Matrix

The following are differences between this example table and the example Requirements Verification Matrix in the NASA SE Handbook, Table D-1:

- “Document”, “Paragraph”, “Phase”, “Performing Organization”, “Acceptance Requirement?” are not included in Figure 8.16-1.
- “Artifact for Verification” was added to Figure 8.16-1.
- “Facility or Lab” information can be captured via the parts of the V&V Configuration element (see the V&V Configuration bdd in section 8.19).

Table D-1 Requirements Verification Matrix

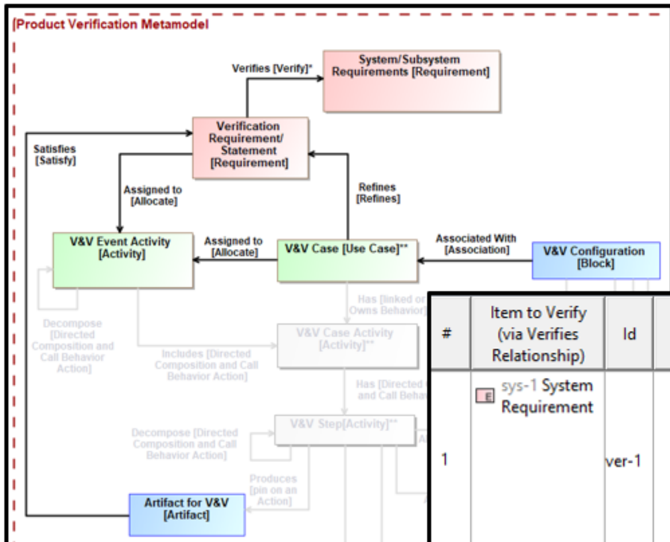
Require-ment No.	Document	Paragraph	Shall Statement	Verification Success Criteria	Verification Method	Facility or Lab	Phase[a]	Acceptance Require-ment?	Preflight Accept-ance?	Performing Organization	Results
Unique identifier or each requirement	*Document number the requirement is contained within*	*Paragraph number of the requirement*	*Text (within reason) of the requirement, i.e., the “shall”*	*Success criteria for the requirement*	*Verification method for the requirement (analysis, inspection, demonstration, test)*	*Facility or laboratory used to perform the verification and validation.*	*Phase in which the verification and validation will be performed*	*Indicate whether this requirement is also verified during initial acceptance testing of each unit.*	*. Indicate whether this requirement is also verified during any pre-flight or recurring acceptance testing of each unit*	*Organization responsible for performing the verification*	*Indicate docu-ments that contain the objective evidence that require - ment was satisfied*
P-1	xxx	3.2.1.1 Capability: Support	System X shall provide a	1. System X locks to forward link	Test	xxx	5	Yes	No	xxx	TPS xxxx

Figure 8.16-2 Sample Requirements Verification Matrix from the NASA SE Handbook[12]

[12] NASA/SP-2016-6105, Revision 2, Table D-1

8.17 Table for Verification Requirements Matrix/ Verification Compliance Spreadsheet (VCS)

A table of verification requirements/ statements and their attributes and relationships are shown in **Figure 8.17-1**, **Verification Requirements Matrix/ Verification Compliance Spreadsheet (Right); Metamodel (Left)**. The metamodel portion for verification from section 7, **Figure 7-2**, is shown on the left and a sample verification requirements matrix or content for a verification compliance spreadsheet is shown on the right. Section 8.15 shows a requirement diagram of the verification requirements and traceability depicted in this table.



#	Item to Verify (via Verifies Relationship)	Id	Name	Verify Method	Text	Success Criteria	Refined By	Artifact	Verification Event	Documentation/ Results
1	sys-1 System Requirement	ver-1	Verification Requirement 1	Test	Measure the amount of power between x and y and see how long it takes to reach a steady state.	Steady State is reached by specfication in requirement	Power Test Case 01 Power Test Case 02	Verification Artifact 1 - Power Meter Read Out	Power Test Event	Closure Status: Closure Date: Closure Notes: Description of how the artifacts met the success criteria. And non-conformance findings and statuses
2	subsys-1 Subsystem Requirement 1	ver-2	Verification Requirement 2	Inspection	Conduct a visual inspection for...	Inspection shows system is in compliance with stated requirement		Verification artifact 2 - Subsys-1 Inspection Photo	Inspection Event	
3	comp-1 Component Requirement 1	ver-3	Verification Requirement 3	Analysis	An analysis shall be performed to record wattage output related to the nom and off-nom scenarios and ensure the maximum power is sufficient.	Analysis shows that system does not exceed 50 watts		Verification artifact 3 - power analysis report	Power Analysis Event	

Figure 8.17-1 Verification Requirements Matrix/ Verification Compliance Spreadsheet (Right); Metamodel (Left)

8.18 Use Case (uc) Diagram of V&V Cases

An example use case diagram of V&V cases is shown in **Figure 8.18-1**, **V&V Case Metamodel (Top); Use Case Diagram of V&V Case (Bottom)**. Use case diagrams describe the functions of a system and the interactions between those functions and system actors or elements. For verification and validation, the use case is capturing the V&V case that owns a V&V activity (see section 8.20 for a V&V activity description). The metamodel portion from the V&V metamodel from section 7, **Figure 7-2**, is shown on top. A sample SysML® use case diagram is shown on the bottom.

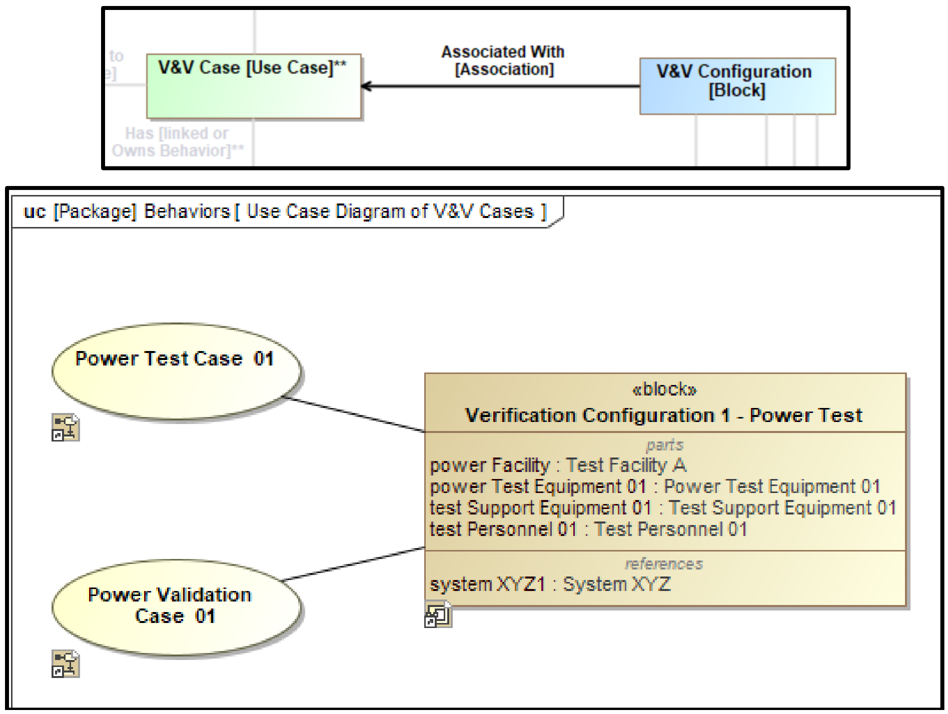


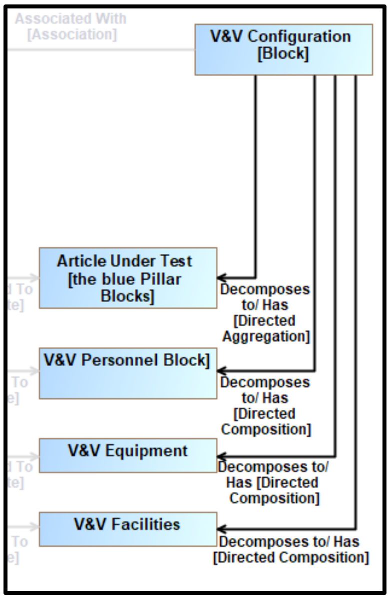
Figure 8.18-1 V&V Case Metamodel (Top); Use Case Diagram of V&V Case (Bottom)

Notes:

- In this example the same V&V configuration (*Verification Configuration 1 – Power Test*) is being used in a Verification Case and a Validation Case example (for more details on the Validation case see section 8.22).
- Actors can represent roles played by human users, external hardware, or any other subjects. In this example the verification configuration represents the actor. An example bdd for the verification configuration block is shown in section 8.19.

8.19 V&V Configuration Decomposition Block Definition Diagram (bdd)

An example V&V configuration decomposition bdd is shown in **Figure 8.19-1**, **Verification Configuration Decomposition Metamodel (Left); bdd (Right)**. The metamodel portion of the V&V configuration decomposition from section 7, **Figure 7-2**, is shown on the left. A sample SysML® bdd of a verification configuration decomposition is shown on the right. Note, this example configuration shows the system block as a reference property and the verification equipment and personnel and support facilities as part properties.



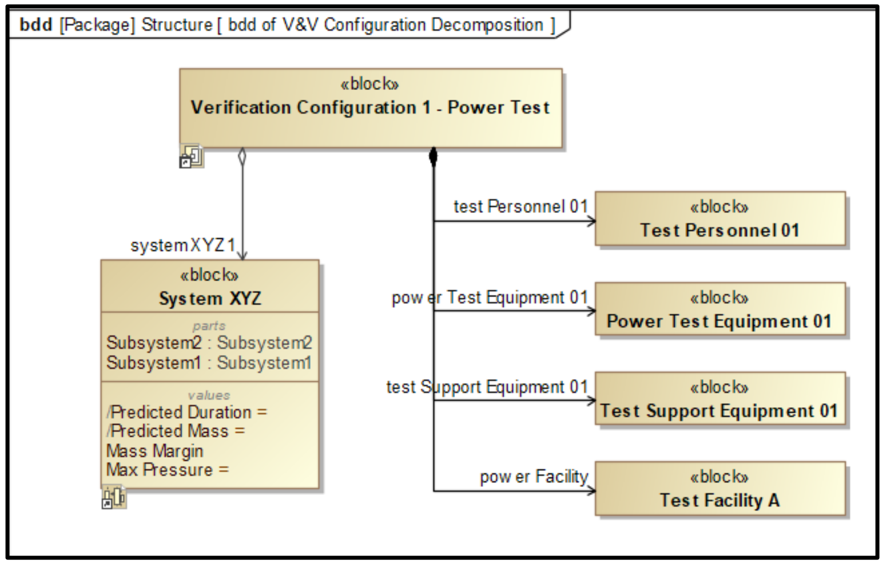


Figure 8.19-1 Verification Configuration Decomposition Metamodel (Left); bdd (Right)

8.20 Activity Diagram (act) of V&V Case/ Sequencing of the V&V Approach

An example activity diagram of a V&V Case is shown in **Figure 8.20-1**, **V&V Step-Activity Elements Allocated to Structure Elements Metamodel (Top); act of Power Test Case 01 (Bottom)**. In this example, the activity diagram is used to further explain the details of the verification case “Power Test Case 01” shown in sections 8.15 and 8.18. Activities can be captured in activity diagrams showing interactions between activities and allocations to structure elements (see Appendix E). The metamodel portion of V&V Case activity elements and their relationships from section 7, **Figure 7-2**, is shown in the top of **Figure 8.20-1**. A sample SysML® activity diagram using swim lanes to allocate the test case steps to block elements of the verification configuration is shown on the bottom.

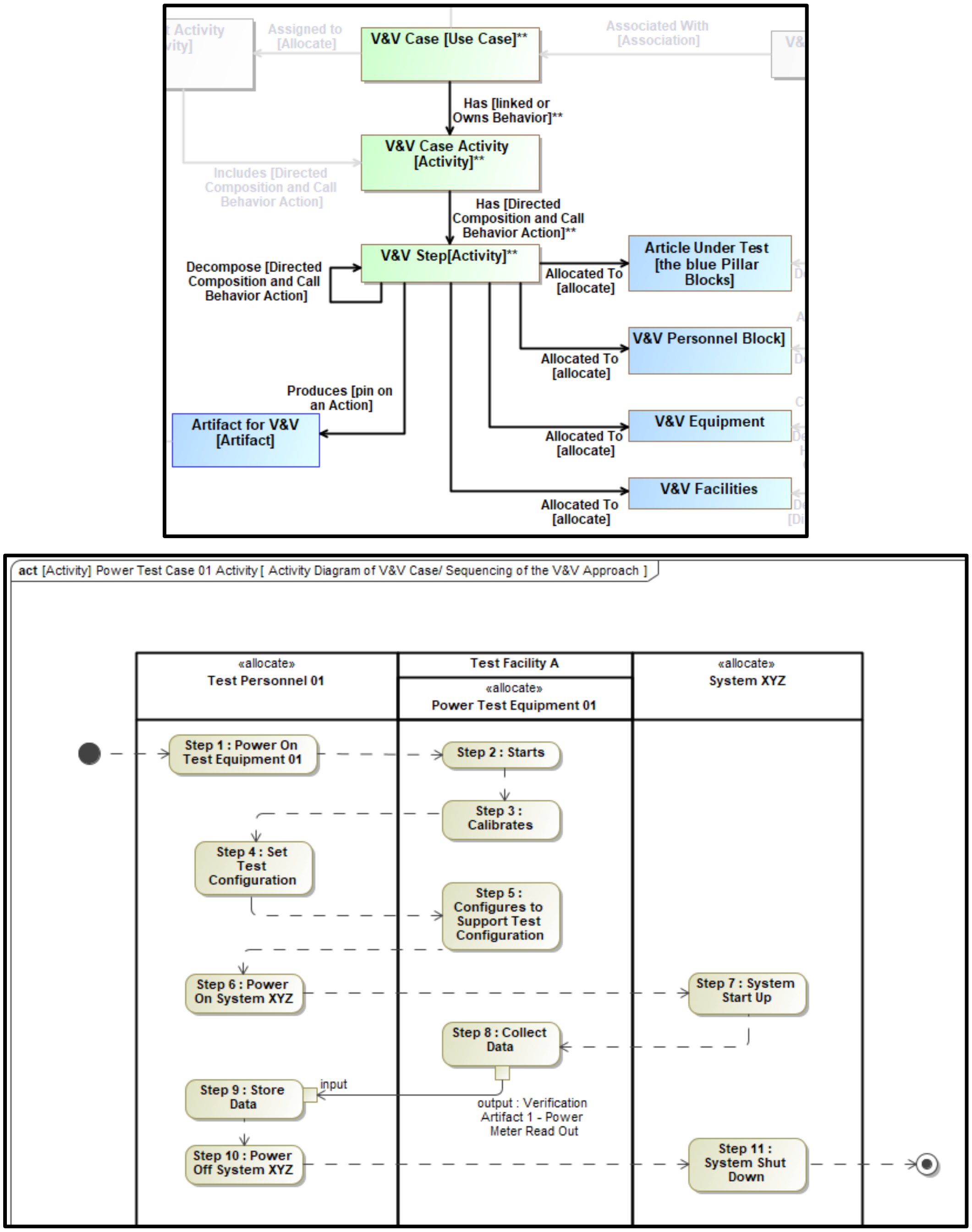


Figure 8.20-1 V&V Step-Activity Elements Allocated to Structure Elements Metamodel (Top); act of Power Test Case 01 (Bottom)

8.21 Requirements Diagram (req) of Validation Requirement and traceability

Figure 8.21-1, Requirements Diagram of Validation Requirement and Traceability, provides an example validation requirements diagram that follows the V&V metamodel from section 7, **Figure 7-2**. The Validation metamodel is shown on the left and an example requirements diagram depicting the related elements is provided on the right. Note the same information can also be depicted in a tabular format (see section 8.22 for an example table).

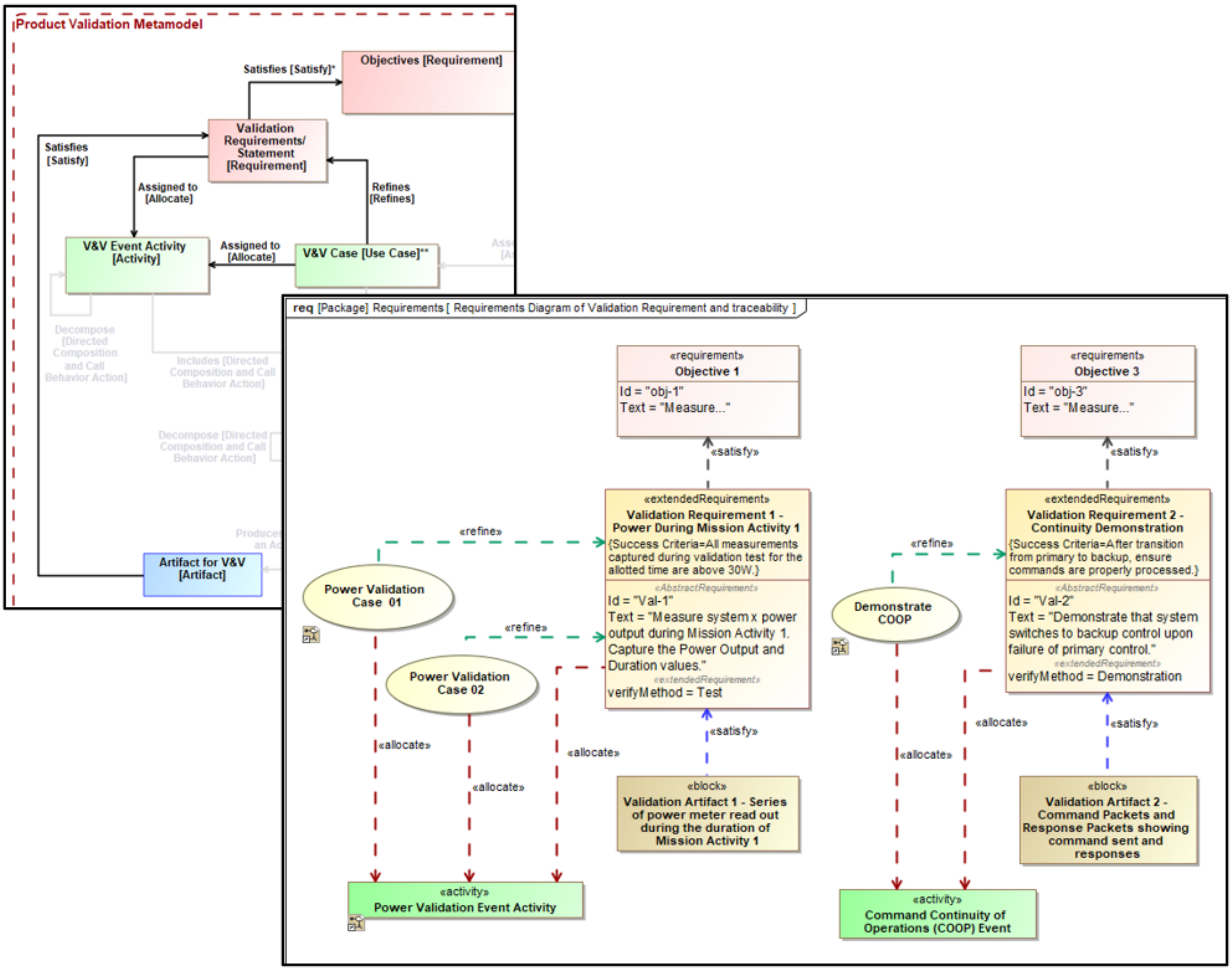
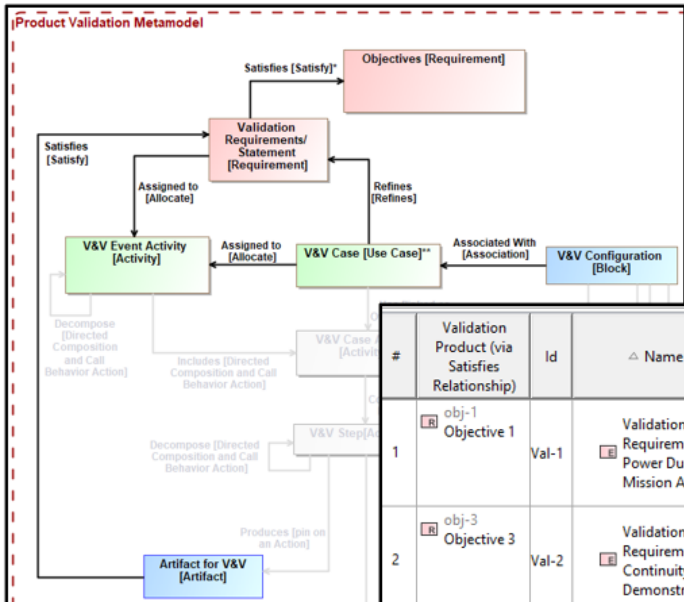


Figure 8.21-1 Requirements Diagram of Validation Requirement and Traceability

> Notes:
> - The success criteria is captured as a constraint relationship on the validation requirements in this example, shown in { } below the *Validation Requirement 1 Power During Mission Activity 1* and *Validation Requirement 2 Continuity Demonstration* elements.
> - Section 8.22 provides a Validation Requirements Matrix/ Validation Compliance Spreadsheet for this example.

8.22 Table for Validation Requirements Matrix/ Validation Compliance Spreadsheet (VCS)

Figure 8.22-1, **Validation Requirements Matrix/ Compliance Spreadsheet (Right); Metamodel (Left)**, provides an example tabular view of the content in the validation requirements diagram in **Figure 8.21-1**. For more information on Validation Requirements Matrix, refer to NASA/SP-2016-6105. The NASA SE Handbook provides a sample Validation Requirements Matrix as seen in **Figure 8.22-2**, **Sample Validation Requirements Matrix from the NASA SE Handbook**.



#	Validation Product (via Satisfies Relationship)	Id	Name	Activity/ Validation Req. Text	Objective/ Success Criteria	Refined By	Validation Method (via the Verify Method)	Validation Artifact	Validation Configuration	Validation Event	Documentation/ Results
1	obj-1 Objective 1	Val-1	Validation Requirement 1 - Power During Mission Activity 1	Measure system x power output during Mission Activity 1. Capture the Power Output and Duration values.	All measurements captured during validation test for the allotted time are above 30W.	Power Validation Case 02 Power Validation Case 01	Test	Validation Artifact 1 - Series of power meter read out during the duration of Mission Activity 1	Verification Configuration 1 - Power Test	Power Validation Event Activity	Closure Status: Closure Date: Closure Notes:
2	obj-3 Objective 3	Val-2	Validation Requirement 2 - Continuity Demonstration	Demonstrate that system switches to backup control upon failure of primary control.	After transition from primary to backup, ensure commands are properly processed.	Demonstrate COOP	Demonstration	Validation Artifact 2 - Command Packets and Response Packets showing command sent and responses		Command Continuity of Operations (COOP) Event	Closure Status: Closure Date: Closure Notes:
3	obj-2 Objective 2	Val-3	Validation Requirement 3				Demonstration				

Figure 8.22-1 Validation Requirements Matrix/ Compliance Spreadsheet (Right); Metamodel (Left)

Comparison to NASA SE Handbook Example Validation Requirements Matrix

The following are differences between this example table and the example Validation Requirements Matrix in the NASA SE Handbook, Table E-1:

- “Phase” and “Performing Organization” are not included in Figure 8.22-1.
- “Validation Cases”, “Validation Artifact”, and “Validation Event” are added to Figure 8.22-1.
- “Facility or Lab” information can be captured via the parts of the V&V Configuration element (see the V&V Configuration bdd in Section 8.19).

Table E-1 Validation Requirements Matrix

Validation Product #	Activity	Objective	Validation Method	Facility or Lab	Phase	Performing Organization	Results
Unique identifier for validation product	*Describe evaluation by the customer/ sponsor that will be performed*	*What is to be accomplished by the customer/ sponsor evaluation*	*Validation method for the requirement (analysis, inspection, demonstration, or test)*	*Facility or laboratory used to perform the validation*	*Phase in which the verification/ validation will be performed[a]*	*Organization responsible for coordinating the validation activity*	*Indicate the objective evidence that validation activity occurred*
1	Customer/ sponsor will evaluate the candidate displays	1. Ensure legibility is acceptable 2. Ensure overall appearance is acceptable	Test	xxx	Phase A	xxx	TPS 123456

a. Example: (1) during product selection process, (2) prior to final product selection (if COTS) or prior to PDR, (3) prior to CDR, (4) during box-level functional, (5) during system-level functional, (6) during end-to-end functional, (7) during integrated vehicle functional, (8) during on-orbit functional.

Figure 8.22-2 Sample Validation Requirements Matrix from the NASA SE Handbook[13]

[13] NASA/SP-2016-6105, Revision 2, Table E-1

9. GENERATING DIAGRAMS AND TABLES FROM THE MODEL TO SUPPORT SYSTEMS ENGINEERING PRODUCTS

This section provides a list of diagrams and tables that can be used to support the NPR 7123.1 systems engineering work products related to Stakeholder Expectations Definition, Technical Requirements Definition, Verification, and Validation SE processes. Once the system model is set up and populated, diagrams and tables can be extracted from the model to visualize, communicate, and deliver data, information, and knowledge to the stakeholders; support technical reviews; and support informed management decisions for progressing to the next life-cycle phase. Section 8 provides examples of a subset of these diagrams and tables. Diagrams and table views can be extracted from the model, either manually, or through third-party tools. They can be used to populate report templates, exported to webpages or other model viewing tools, or used directly within a system model tool allowing navigation between diagrams and tables within the model tool.

The work products covered in this section are Stakeholder Identification and Expectations Definition, ConOps, MOE Definition, Technical Requirements, MOP, TPM, and V&V Requirements, Planning, Results and Reports (see section 4.1.1 for reference).

9.1 Generating SysML® Diagrams and Tables for Stakeholder Identification and Expectations Definition

Example SysML® diagrams and tables to support stakeholder identification and expectations definition SE work product are depicted in **Figure 9.1-1**, **Diagrams and Tables to Support Stakeholder Identification and Expectations**.

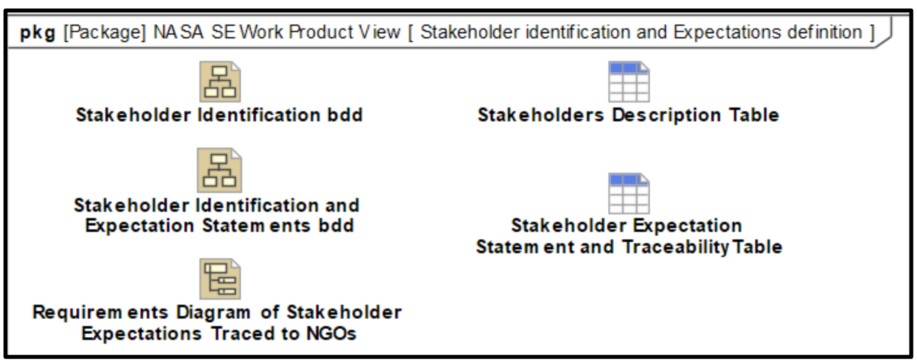


Figure 9.1-1 Diagrams and Tables to Support Stakeholder Identification and Expectations

The diagrams and tables depicted in **Figure 9.1-1** are listed in the following table:

Table 9.1-1 Diagrams and Tables to Support Stakeholder Identification and Expectations

List of Diagrams and Tables	Section 8 Reference
Stakeholder Identification bdd	
Stakeholders Description Table	Section 8.1
Stakeholder Identification and Expectation Statements bdd	Section 8.1
Stakeholder Expectation Statement and Traceability Table	Section 8.1
Requirements Diagrams of Stakeholder Expectations Traced to NGOs	

9.2 Generating SysML® Diagrams and Tables for Concept of Operations (ConOps)

ConOps, describes the overall high-level concept of how the system will be used to meet stakeholder expectations, usually in a time-sequenced manner. [14] Example SysML® diagrams and tables to support a ConOps product are depicted in **Figure 9.2-1**, **Diagrams and Tables to Support Concept of Operations (ConOps) Product**. (Note: The diagrams and tables selected by a program/project will depend on a program/project's ConOps development level. For example, the System Context bdd and ibd may be sufficient for a technical review in one case. If a program/project is further along, the bdd of systems and subsystems may be included.)

> See Appendix F for a sample ConOps document template populated with MBSE diagrams and tables.

Figure 9.2-1 Diagrams and Tables to Support Concept of Operations (ConOps) Product

[14] NPR 7123.1D, Appendix A. Definitions

The diagrams and tables depicted in **Figure 9.2-1** are listed in the following table:

Table 9.2-1 Diagrams and Tables to Support Concept of Operations (ConOps) Product

List of Diagrams and Tables	Section 8 Reference	Appendix F Reference
System Context bdd Diagram	Section 8.3	Section 1.2; Section 3.2
System Context ibd Diagram	Section 8.4	Section 1.2; Section 3.3
Requirements Diagram of NGOs	Section 8.2	Section 3.1
Requirements Table of NGOs	Section 8.17	Section 3.1
bdd of Mission Phase and Activity Decomposition	Section 8.16	Section 3.5
Activity Diagram of Mission Phases		Section 3.5
System Use Cases	Section 8.5	Section 3.5; Section 6.0
Table of Actors traced to Activities and Use Cases		Section 3.2; Section 3.5
Mission Phase 1 Activity Diagram to Support Use Cases	Section 8.6	Section 6.0
Table of Activities with Allocated Elements		Section 3.5
System XYZ State Machine of Modes		Section 3.4
Table of Modes and Descriptions		Section 3.4
Functional Decomposition of Activities via a bdd	Section 8.9	
Table of Elements with Descriptions and Allocated Activities		Section 3.2
bdd of Structural Decomposition	Section 8.7	
ibd of Structure Interconnections	Section 8.8	Section 3.3
Table of System Context Interfaces		Section 3.3
Table of Interface Items and Descriptions		Section 3.3
Subsystem 1 State Machine		Section 6.0
Sequence Diagram of Subsystem1 Interactions		Section 6.0
bdd of Blocks with Values Related to MOEs and/or MOPs		Section 3.2
Requirements Table with Objectives and Refined MOEs		
Requirements Table with Requirements and Refined TPMs		
Requirements Table of Validation Requirements/Statement		

9.3 Generating SysML® Diagrams and Tables for Measures of Effectiveness (MOE) Definition

MOE is a measure by which a stakeholder's expectations will be judged in assessing satisfaction with products or systems produced and delivered in accordance with the associated technical effort. An MOE is typically qualitative in nature.[15]

Example SysML® diagrams and tables to support MOE definition are depicted in **Figure 9.3-1**, **Diagrams and Tables to Support MOE Definition**.

[15] NPR 7123.1D, Appendix A. Definitions

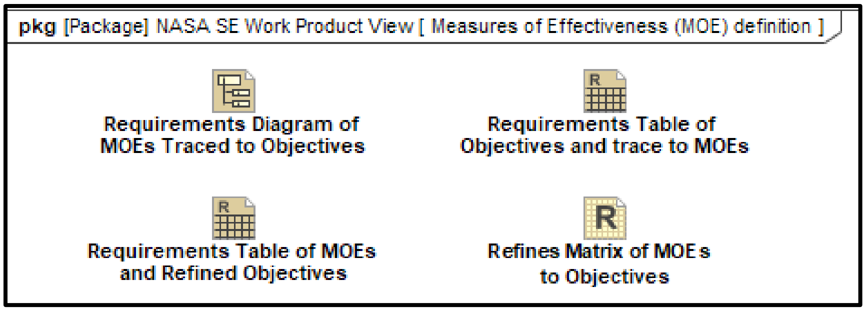


Figure 9.3-1 Diagrams and Tables to Support MOE Definition

The diagrams and tables depicted in **Figure 9.3-1** are listed in the following table:

Table 9.3-1 Diagrams and Tables to Support MOE Definition

List of Diagrams and Tables	Section 8 Reference
Requirements Diagram of MOEs Traced to Objectives	Section 8.12
Requirements Table with Objectives and trace to MOEs	Section 8.12
Requirements Table of MOEs and Refined Objectives	
Refines Matrix of Objectives to MOEs	

9.4 Generating SysML® Diagrams and Tables for Technical Requirements

Example SysML® diagrams and tables to support requirements products are depicted in **Figure 9.4-1**, **Diagrams and Tables to Support Technical Requirements**.

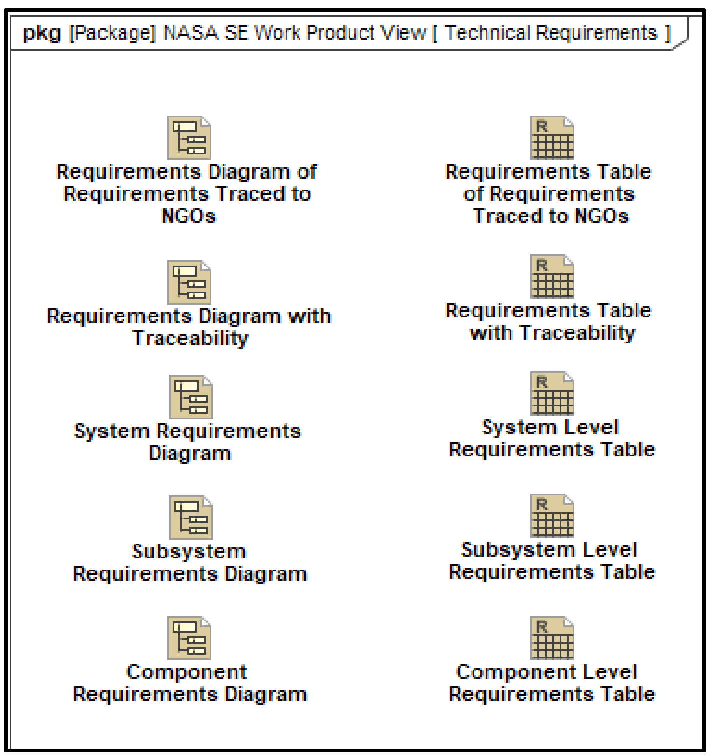


Figure 9.4-1 Diagrams and Tables to Support Technical Requirements

The diagrams and tables depicted in **Figure 9.4-1** are listed in the following table:

Table 9.4-1 Diagrams and Tables to Support Technical Requirements

List of Diagrams and Tables	Section 8 Reference
Requirements Diagram of Requirements Traced to NGOs	
Requirements Table of Requirements Traced to NGOs	
Requirements Diagram with Traceability	Section 8.10
Requirements Table with Traceability	Section 8.11
System Requirements Diagram	
System Level Requirements Table	
Subsystem Requirements Diagram	
Subsystem Level Requirements Table	
Component Requirements Diagram	
Component Level Requirements Table	

9.5 Generating SysML® Diagrams and Tables for Measures of Performance (MOP)

MOP is a quantitative measure that, when met by the design solution, will help ensure that the MOE for a product or system will be satisfied.[16]

In the metamodel, the MOP uses a requirement element to capture the quantitative measure to be met by the design solution. Then, the “satisfies” relationship between model elements (like the TPM value property) is the trace used for the “met” part of this definition description. The MOP has a trace to the MOE.

Example SysML® diagrams and tables to support MOP definition are depicted in **Figure 9.5-1, Diagrams and Tables to Support MOP Definition**.

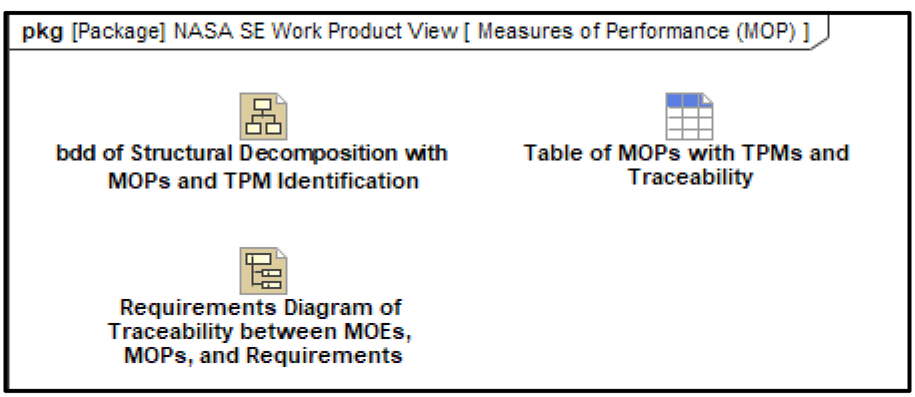


Figure 9.5-1 Diagrams and Tables to Support MOP Definition

The diagrams and tables depicted in **Figure 9.5-1** are listed in the following table:

[16] NPR 7123.1D, Appendix A. Definitions

Table 9.5-1 Diagrams and Tables to Support MOP Definition

List of Diagrams and Tables	Section 8 Reference
bdd of Structural Decomposition with MOPs and TPM Identification	Section 8.13
Table of MOPs with TPMs and Traceability	Section 8.13
Requirements Diagram of Traceability between MOEs, MOPs, and Requirements	Section 8.12

9.6 Generating SysML® Diagrams and Tables for Technical Performance Measures (TPM)

TPM is the set of performance measures that are monitored by comparing the current actual achievement of the parameters with that anticipated at the current time and on future dates. Technical performance measures are typically selected from the defined set of Measures of Performance (MOPs).[17]

Example SysML® diagrams and tables to support TPM definition are depicted in **Figure 9.6-1, Diagrams and Tables to Support TPM Definition**.

Note: Tools can be used to determine TPM calculations based on inserted values (for specific NASA resources for TPM calculation techniques visit the NASA MBSE Community of Practice website at https://nen.nasa.gov/web/mbse/ (Search: TPM).

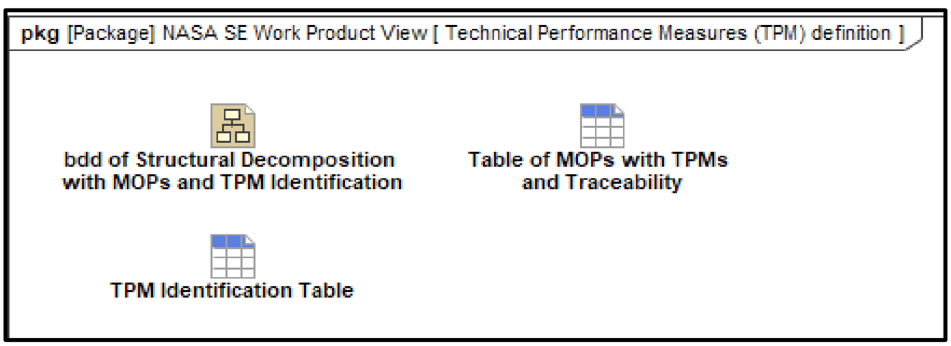


Figure 9.6-1 Diagrams and Tables to Support TPM Definition

The diagrams and tables depicted in **Figure 9.6-1** are listed in the following table:

Table 9.6-1 Diagrams and Tables to Support TPM Definition

List of Diagrams and Tables	Section 8 Reference
bdd of Structural Decomposition with MOPs and TPM Identification	Section 8.13
Table of MOPs with TPMs and Traceability	Section 8.13
TPM Identification Table	

[17] NPR 7123.1D, Appendix A. Definitions

9.7 Generating SysML® Diagrams and Tables for Verification and Validation (V&V)

Example SysML® diagrams and tables to support Verification & Validation Requirements, Planning, Results and Reports is depicted in **Figure 9.7-1**, **Diagrams and Tables to Support Verification and Validation (V&V) Products**.

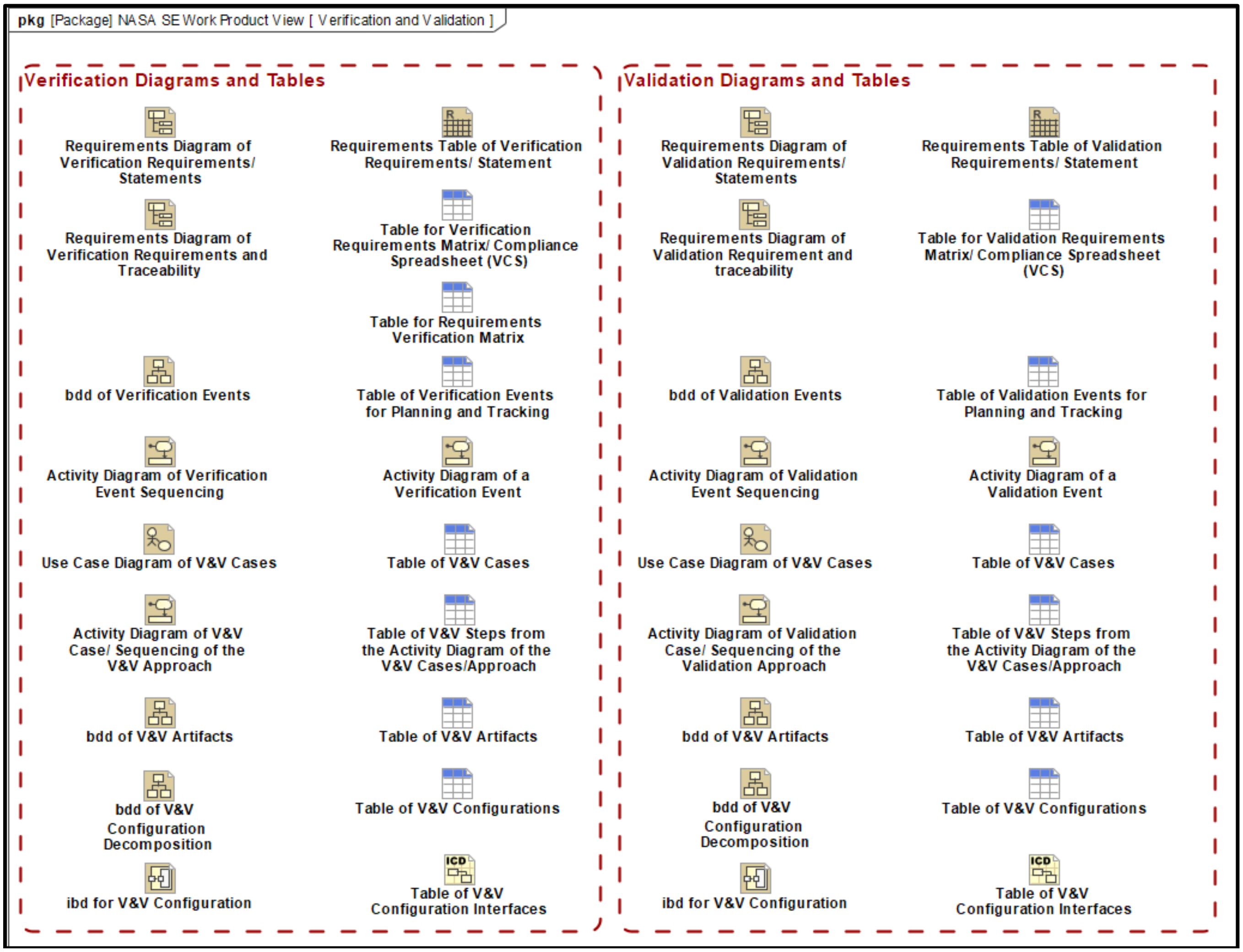


Figure 9.7-1 Diagrams and Tables to Support Verification and Validation (V&V) Products

> Note: The figure above shows that some diagrams and tables can be used for both Verification and Validation processes. The table below indicates which of these diagrams and tables are shown in which view. In these shared cases, the modeler may choose to create separate diagrams and tables for each process as needed by the project.

The diagrams and tables depicted in **Figure 9.7-1** are listed in the following table:

Table 9.7-1 Diagrams and Tables to Support V&V Products

List of Diagrams and Tables	Verification View	Validation View	Section 8 Reference
Requirements Diagram of Verification Requirements/ Statements	X		Section 8.14
Requirements Table of Verification Requirements/ Statement	X		
Table for Requirements Diagram of Verification Requirements and Traceability	X		Section 8.15
Table for Verification Requirements Matrix/ Compliance Spreadsheet (VCS)	X		Section 8.17
Requirements Verification Matrix	X		Section 8.16
bdd of Verification Events	X		
Table of Verification Events for Planning and Tracking	X		
Activity Diagram of Verification Event Sequencing	X		
Activity Diagram of a Verification Event	X		
Requirements Diagram of Validation Requirements/ Statements		X	
Requirements Table of Validation Requirements/ Statement		X	
Requirements Diagram of Validation Requirement and traceability		X	
Table for Validation Requirements Matrix/ Compliance Spreadsheet (VCS)		X	
bdd of Validation Events		X	
Table of Validation Events for Planning and Tracking		X	
Activity Diagram of Validation Event Sequencing		X	
Activity Diagram of a Validation Event		X	
Use Case Diagram of V&V Cases	X	X	Section 8.18
Table of V&V Cases	X	X	
Activity Diagram of V&V Case/ Sequencing of the V&V Approach	X	X	Section 8.20
Table of V&V Steps from the Activity Diagram of the V&V Cases/Approach	X	X	
bdd of V&V Artifacts	X	X	
Table of V&V Artifacts	X	X	
bdd of V&V Configuration Decomposition	X	X	Section 8.19
Table of V&V Configurations	X	X	
ibd for V&V Configuration	X	X	
Table of V&V Configuration Interfaces	X	X	

APPENDIX A: NASA SE COMPETENCY MODEL

A.1 Purpose

This appendix provides a description of each of the common technical processes depicted in **Figure 4.1-1**, **NASA Systems Engineering Engine**.

A.2 NASA SE Competency Model

NPR 7123.1 details three sets of common technical processes: system design, product realization, and technical management. The processes in each set and their descriptions are provided in **Figure A.2-1, NASA Systems Engineering Competency Model**.

Competency Area	Competency	Description
SE 1.0 System Design	SE 1.1 Stakeholder Expectation Definition & Management	Eliciting and defining use cases, scenarios, concept of operations and stakeholder expectations. This includes identifying stakeholders, establishing support strategies, establishing a set of Measures of Effectiveness (MOEs), validating stakeholder expectation statements, and obtaining commitments from the customer and other stakeholders, as well as using the baselined stakeholder expectations for product validation during product realization
	SE 1.2 Technical Requirements Definition	Transforming the baseline stakeholder expectations into unique, quantitative, and measurable technical requirements expressed as "shall" statements that can be used for defining the design solution. This includes analyzing the scope of the technical problems to be solved, defining constraints affecting the designs, defining the performance requirements, validating the resulting technical requirement statements, defining the Measures of Performance (MOPs) for each MOE, and defining appropriate Technical Performance Measures (TPMs) by which technical progress will be assessed.
	SE 1.3 Logical Decomposition	Transforming the defined set of technical requirements into a set of logical decomposition models and their associated set of derived technical requirements for lower levels of the system, and for input to the design solution efforts. This includes decomposing and analyzing by function, time, behavior, data flow, object, and other models. It also includes allocating requirements to these decomposition models, resolving conflicts between derived requirements as revealed by the models, defining a system architecture for establishing the levels of allocation, and validating the derived technical requirements.
	SE 1.4 Design Solution Definition	Translating the decomposition models and derived requirements into one or more design solutions, and using the Decision Analysis process to analyze each alternative and for selecting a preferred alternative that will satisfy the technical requirements. A full technical data package is developed describing the selected solution. This includes generating a full design description for the selected solution; developing a set of 'make-to,' 'buy-to,' 'reuse-to,' specifications; and initiating the development or acquisition of system products and enabling products.
SE 2.0 Product realization	SE 2.1 Product Implementation	Generating a specific product through buying, making, or reusing so as to satisfy the design requirements. This includes preparing the implementation strategy; building or coding the produce; reviewing vendor technical information; inspecting delivered, built, or reused products; and preparing product support documentation for integration.
	SE 2.2 Product Integration	Assembling and integrating lower-level validated end products into the desired end product of the higher-level product. This includes preparing the product integration strategy, performing detailed planning, obtaining products to integrate, confirming that the products are ready for integration, preparing the integration environment, and preparing product support documentation.
	SE 2.3 Product Verification	Proving the end product conforms to its requirements. This includes preparing for the verification efforts, analyzing the outcomes of verification (including identifying anomalies and establishing recommended corrective actions), and preparing a product verification report providing the evidence of product conformance with the applicable requirements.
	SE 2.4 Product Validation	Confirming that a verified end product satisfies the stakeholder expectations for its intended use when placed in its intended environment and ensuring that any anomalies discovered during validation are appropriately resolved prior to product transition. This includes preparing to conduct product validation, performing the product validation, analyzing the results of validation (including identifying anomalies and establishing recommended corrective actions), and preparing a product validation report providing the evidence of product conformance with the stakeholder expectations baseline.
	SE 2.5 Product Transition	Transitioning the verified and validated product to the customer at the next level in the system structure. This includes preparing to conduct product transition, evaluating the product and enabling product readiness for product transition, preparing the product for transition (including handling, storing, and shipping preparation), preparing sites, and generating required documentation to accompany the product
SE 3.0 Technical Management	SE 3.1 Technical Planning	Planning for the application and management of each common technical process, as well as identifying, defining, and planning the technical effort necessary to meet project objectives. This includes preparing or updating a planning strategy for each of the technical processes, and determining deliverable work products from technical efforts; identifying technical reporting requirements; identifying entry and success criteria for technical reviews; identifying product and process measures to be used; identifying critical technical events; defining cross domain interoperability and collaboration needs; defining the data management approach; identifying the technical risks to be addressed in the planning effort; identifying tools and engineering methods to be employed; and defining the approach to acquire and maintain technical expertise needed. This also includes preparing the Systems Engineering Management Plan (SEMP) and other technical plans; obtaining stakeholder commitments to the technical plans; and issuing authorized technical work directives to implement the technical work
	SE 3.2 Requirements Management	Managing the product requirements, including providing bidirectional traceability, and managing changes to establish requirement baselines over the life cycle of the system products. This includes preparing or updating a strategy for requirements management; selecting an appropriate requirements management tool; training technical team members in established requirement management procedures; conducting expectation and requirements traceability audits; managing expectation and requirement changes; and communicating expectation and requirement change information
	SE 3.3 Interface Management	Establishing and using formal interface management to maintain internal and external interface definition and compliance among the end products and enabling products. This includes preparing interface management procedures, identifying interfaces, generating and maintaining interface documentation, managing changes to interfaces, disseminating interface information, and conducting interface control
	SE 3.4 Technical Risk Management	Examining on a continual basis the risks of technical deviations from the plans, and identifying potential technical problems before they occur. Planning, invoking, and performing risk-handling activities as needed across the life of the product or project to mitigate impacts on meeting technical objectives. This includes developing the strategy for technical risk management, identifying technical risks, and conducting technical risk assessment; preparing for technical risk mitigation, monitoring the status of each technical risk, and implementing technical risk mitigation and contingency action plans when applicable thresholds have been triggered.

Figure A.2-1 NASA Systems Engineering Competency Model[18]

[18] NASA/SP-2016-6105, Revision 2, Table 2.7-1

APPENDIX B: MODELING METHODOLOGY AND FRAMEWORK BACKGROUND

B.1 Purpose

This appendix provides additional information on OOSEM methodology and the MBSE Grid framework and how they relate to the NASA modeling methodology and framework described in sections 4.3.2 and 4.3.3 in this Handbook.

B.2 Modeling Methodology Background

OOSEM is a systems-level development method that combines object-oriented concepts with traditional systems engineering practices. **Figure B.1-1, OOSEM System Development Workflow**, shows the top-level OOSEM process in blue and secondary level processes in white. The OOSEM System Development Workflow shows the Update Modeling Plan and Setup Model process steps that were leveraged in section 4.3.2.

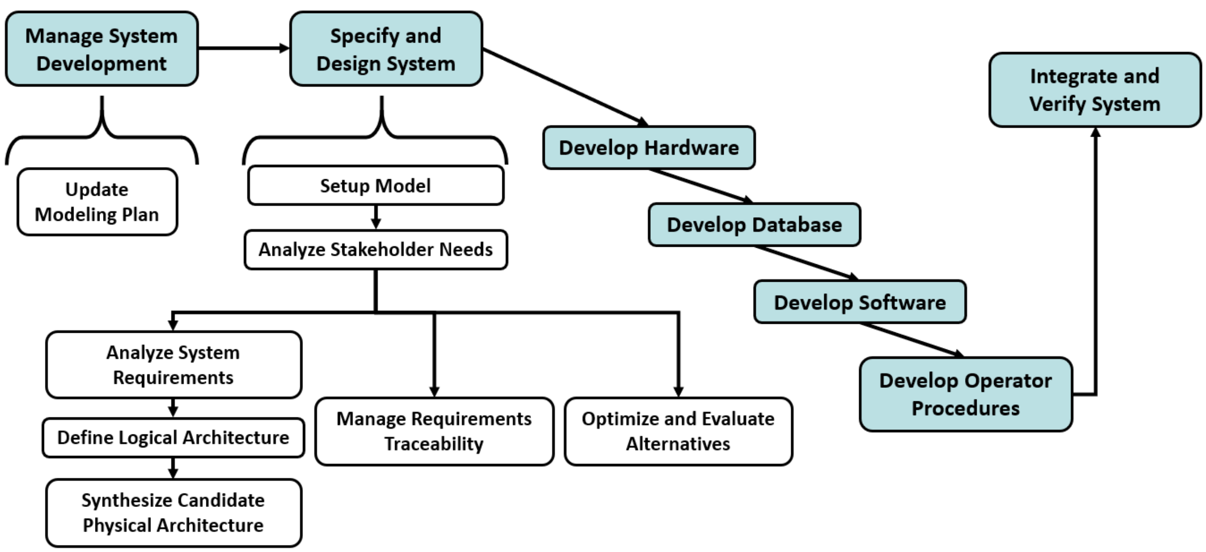


Figure B.1-1 OOSEM System Development Workflow[19]

Note: **Figure 4.3-4,** shows the System Design Process steps from the NASA SE Handbook; these steps are similar to the steps in the 'Specify and Design System' process in the OOSEM workflow in **Figure B.1-1**.

B.3 Modeling Framework Background

The modeling framework in this Handbook leverages the MBSE Grid (shown in **Figure B.2-1, MBSE Grid Framework and Traceability**) and tailors it to the NASA SE Engine. The MBSE

[19] Adapted from the OOSEM Process Baseline model (https://www.incose.org/communities/working-groups-initiatives/object-oriented-se-method) (see OOSEM Process Baseline (1/2020) model)

Grid depicts the project life-cycle phase in rows and the Modeling Pillars of SysML® in columns. The content in the cross-sections represent the metamodel which captures the system modeling elements and their relationships thru multiple SysML diagrams.

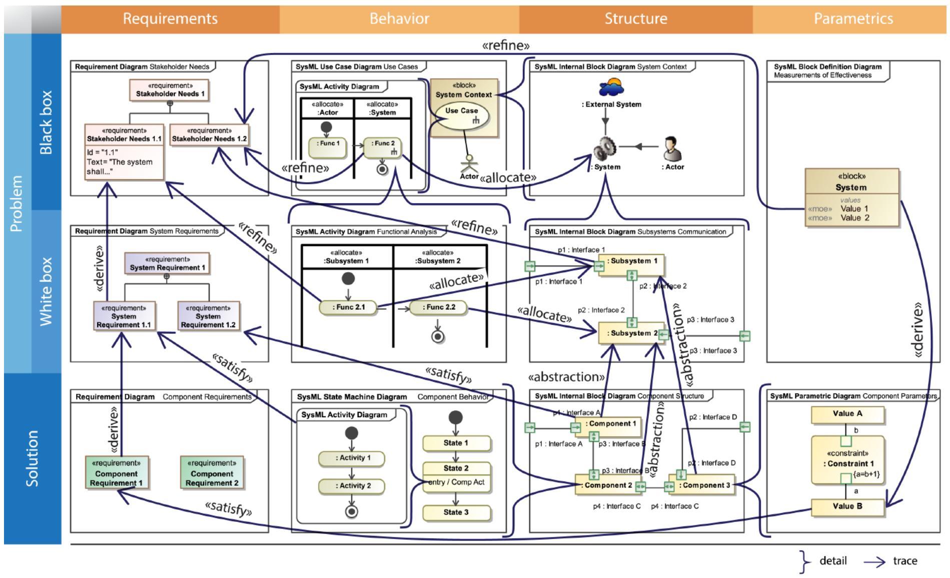


Figure B.2-1 MBSE Grid Framework and Traceability[20]

Figure B.2-2, **MBSE Grid Metamodel**, captures the same model element types and relationships from **Figure B.2-1** in the same format used to capture the metamodel of the NASA SE elements and relationships in section 7, **Figure 7-1** and **Figure 7-2**.

[20] Morkevicius, A.; Aleksandraviciene, A.; Mazeika, D.; Bisikirskiene, L.; & Strolia, Z. (2017). "MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems." INCOSE International Symposium (Vol. 27, No. 1, pp. 136-150). (https://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2017.00350.x)

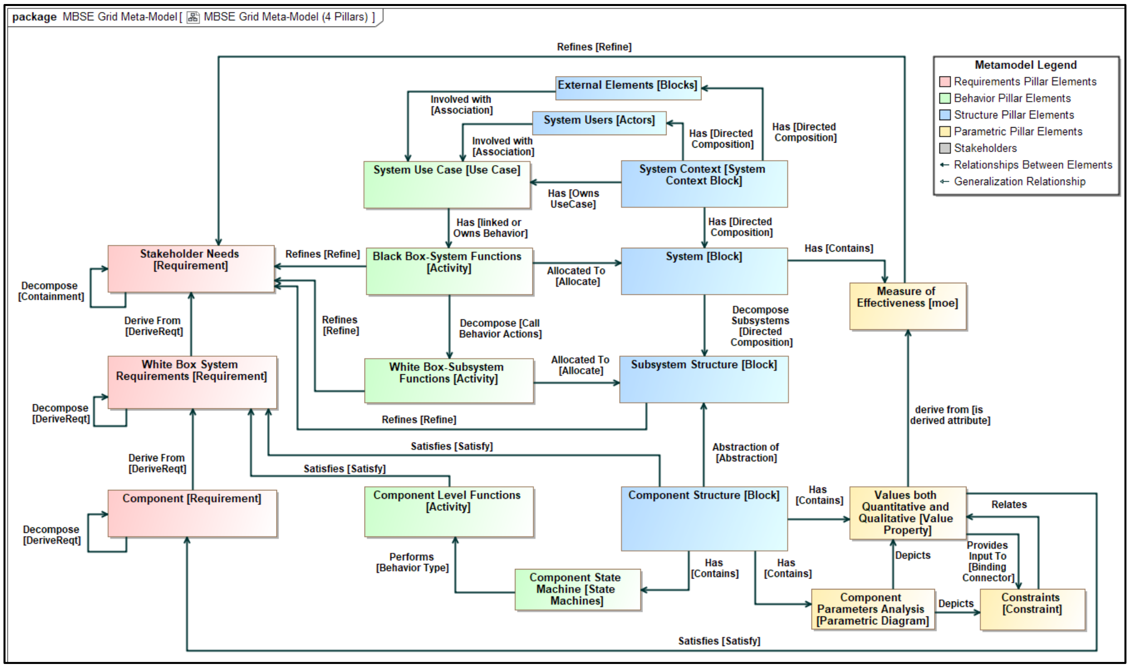


Figure B.2-2 MBSE Grid Metamodel

Differences between the MBSE Grid Framework metamodel and the metamodel based on NASA SE elements and relationships in section 7, **Figure 7-1** and **Figure 7-2**, include:

- Explicit call-out of goals and objectives traced from the stakeholder needs.
- Addition of mission-level behavior and structure elements.
- Addition of validation requirements/statements that trace to objectives.
- Addition of verification requirements/statements that trace to technical requirements.
- Updates to the refines relationships to trace between requirements and behavior elements at the same level of decomposition.
- Addition of allocations between requirements and structure pillars.
- Addition of a decompose relationship to the component level behavior and structure elements from the level above it.
- Addition of Measure of Performances (MOP) and Technical Performance Measures (TPM) value property types.

In the MBSE Grid, project life-cycle phases are divided into two horizontal sections: the “problem” defines and provides an understanding of the problem, and the “solution” provides at least one or more design alternatives to the identified problem. The “problem” section is divided into two rows: “black box” which represents the conceptual representation and “white box” which represents the technical description.

Relating the MBSE Grid to the NASA SE Engine, the life-cycle phases in the MBSE Grid shown in **Figure B.2-1** can be represented by processes 1 through 9 in the NASA SE Engine depicted in **Figure 4.3-5**.

The four System Design Processes map to the MBSE Grid life-cycle phases as follows: the Stakeholder Expectation Definition represents the row for the Black Box; the Technical Requirements Definition and Logical Decomposition map to the White Box; and the Design Solution Definition maps to the Solution layer. Additional rows can be added for the product realization process steps. **Figure 4.3-6** shows the MBSE Grid with NASA SE Processes depicted in the rows and the 4 SysML pillars represented Columns.

The MBSE Grid can be used to organize SysML diagrams and tables from the model within the cross-Section (see **Figure B.2-3**, **MBSE Grid with Diagram Call-Outs**).[20]

			Pillar			
Layer of Abstraction			Requirements	Behavior	Structure	Parametrics
	Problem	Black box	**Stakeholder Needs:** • Requirements diagram • Requirements table	**Use Cases:** 1. Use Case diagram 2. Activity diagram	**System Context:** • Internal block diagram	**Measurements of Effectiveness:** • Block definition diagram
		White box	**System Requirements:** • Requirements diagram • Requirements table	**Functional Analysis:** • Activity diagram	**Logical Subsystems Communication:** 1. Block definition diagram 2. Internal block diagram	**MoEs for Subsystems:** • Block definition diagram
	Solution		**Component Requirements:** • Requirements diagram • Requirements table	**Component Behavior:** • State machine diagram • Activity diagram • Sequence diagram	**Component Structure:** 1. Block definition diagram 2. Internal block diagram	**Component Parameters:** • Parametric diagram

Figure B.2-3 MBSE Grid with Diagram Call-Outs[21]

Appendix C shows an example grid with NASA SE Processes depicted in the rows and the 4 SysML pillars in columns, with the diagrams and tables from this handbook in the cross-sections.

[21] Morkevicius, A.; Aleksandraviciene, A.; Mazeika, D.; Bisikirskiene, L.; & Strolia, Z. (2017). “MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems.” INCOSE International Symposium (Vol. 27, No. 1, pp. 136-150). (https://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2017.00350.x)

APPENDIX C: MODEL CONTENT ORGANIZED IN GRID FORMAT

C.1 Purpose

This appendix provides an example grid with NASA SE processes depicted in the rows and the 4 SysML pillars in columns, with the diagrams and tables from this Handbook in the cross-sections.

C.2 Grid with NASA SE Processes and Handbook Diagram and Table Callouts

Note: **Figure C.2-1, Grid with NASA SE Processes and Handbook Diagram and Table Callouts,** shows some diagram and tables duplicated in multiple rows (for example the Behavior content in the Logical Decomposition and the Design Solution rows). Similar diagrams and tables can be used for multiple processes, with content related to that process.

NASA SYSTEMS ENGINEERING TECHNICAL PROCESSES (1-9)	TECHNICAL PROCESSES	SYSML PILLARS: REQUIREMENTS	SYSML PILLARS: BEHAVIOR	SYSML PILLARS: STRUCTURE	SYSML PILLARS: PARAMETERS
	1. Stakeholder Expectations Definition	**Stakeholder Needs** Stakeholder Identification and Expectation Statements bdd Stakeholder Expectation Statement and TraceabilityTable Requirements Diagram of Stakeholder Expectations Traced to NGOs Requirements Diagram of NGOs Requirements Table of NGOs Requirements Table of Objectives and trace to MOEs	**Use Cases** System Use Cases Table of Actors traced to Activities and Use Cases	**System Context** Stakeholder Identification bdd Stakeholders Description Table System Context bdd Diagram System Context ibd Diagram Table of System Context Interfaces	**Measurements of Effectiveness** Requirements Diagram of MOEs Traced to Objectives Requirements Table of MOEs and Refined Objectives Refines Matrix of MOEs to Objectives bdd of Blocks with Values Related to MOEs and/or MOPs
	2. Technical Requirements Definition 3. Logical Decomposition	**Requirements** Requirements Diagram with Traceability Requirements Table with Traceability System Requirements Diagram System Level Requirements Table Subsystem Requirements Diagram Subsystem Level Requirements Table	**Functional Analysis** bdd of Mission Phase and Activity Decomposition Activity Diagram of Mission Phases Mission Phase 1 Activity Diagram to Support Use Case Functional Decomposition of Activities via a bdd Table of Activities with Allocated Elements	**Logical Architecture and Interfaces** bdd of Structural Decomposition Table of Elements with Descriptions and Allocated Activities ibd of Structure Interconnections Table of Interface Items and Descriptions	**Measurements of Effectiveness and TPMs** bdd of Structural Decomposition with MOPs and TPM Identification TPM Identification Table Table of MOPs with TPMs and Traceability
	4. Design Solution Definition	Component Requirements Diagram Component Level Requirements Table Requirements Table with Requirements and Refined TPMs	**System/ Subsystem/ Component Behavior** Functional Decomposition of Activities via a bdd Table of Activities with Allocated Elements System XYZ State Machine of Modes Table of Modes and Descriptions System XYZ Subsystem 1 State Machine	**System/ Subsystem/ Component Structure and Interfaces** bdd of Structural Decomposition Table of Elements with Descriptions and Allocated Activities ibd of Structure Interconnections Table of Interface Items and Descriptions	**System/ Subsystem/ Component Parameters** bdd of Structural Decomposition with MOPs and TPM Identification bdd of Blocks with Values Related to MOEs and/or MOPs
	5. Product Implementation 6. Product Integration	–	–	–	–
	7. Product Validation 8. Product Verification	**V&V Requirements** Requirements Table of Verification Requirements/ Statement Table for Requirements Verification Matrix Table for Verification Requirements Matrix/ Compliance Spreadsheet (VCS) Requirements Diagram of Verification Requirements and Traceability Requirements Table of Validation Requirements/ Statement Table for Validation Requirements Matrix/ Compliance Spreadsheet (VCS) Requirements Diagram of Validation Requirement and traceability Requirements Diagram of Verification Requirements/ Statements Requirements Diagram of Validation Requirements/ Statements	**V&V Behavior** Use Case Diagram of V&V Cases Activity Diagram of V&V Case/ Sequencing of the V&V Approach Table of V&V Cases Table of V&V Steps from the Activity Diagram of the V&V Cases/Approach bdd of Verification Events Activity Diagram of Verification Event Sequencing Table of Verification Events for Planning and Tracking Activity Diagram of a Verification Event bdd of Validation Events Activity Diagram of Validation Event Sequencing Table of Validation Events for Planning and Tracking Activity Diagram of a Validation Event	**V&V Structure and Interfaces** bdd of V&V Artifacts Table of V&V Artifacts bdd of V&V Configuration Decomposition ibd for V&V Configuration Table of V&V Configuration Interfaces Table of V&V Configurations	**Validated/ Satisfied MoEs and TPMs** –
	9. Product Transition	–	–	–	–

Figure C.2-1 Grid with NASA SE Processes and Handbook Diagram and Table Callouts

APPENDIX D: OTHER MODELING APPROACHES

D.1 Purpose

This appendix provides examples of varying modeling approaches to implement the NASA SE elements and relationships. As mentioned in section 7, the metamodel presented in this Handbook is one approach to modeling in support of the NASA SE Engine. The following describes some of these efforts, providing another approach to requirements modeling, an extension to the behavior modeling, an addition to structure modeling, and an approach for verification analysis.

D.2 Requirements Modeling

The Property-Based Requirement (PBR) modeling approach classes allow for requirements with structure, numerical attributes, and constraints to support requirements analysis.[22] The PBR model element is an extension of the SysML® abstractRequirement, extendedRequirement, and Block[23]. Relating this method to the metamodel in section 7, **Figure 7-1**, all the elements in the Requirements are typed as [PBR Requirement].

D.3 Scenario Modeling

The scenario modeling approach is an extension to behavior modeling. Scenarios can be used to support ConOps development. A Scenario Modeling Context Block is added to serve as a bridge between Activities and State Machines. The Scenario Modeling Context Block uses a directed composition to the system of interest (in the metamodel in section 7, **Figure 7-1**, this refers to any of the elements in the Structure Pillar). It also uses a directed composition relationship to Activities, like those in the Behavior Pillar in **Figure 7-1** that relate to the system of interest. This scenario modeling allows multiple scenarios to be captured, nominal system scenarios, off-nominal, and different level of system composition (top-level system context to low-level component) and can facilitate simulations and additional analysis. **Figure D.3-1, Scenario Modeling Pattern Structure**, shows a sample model of the Scenario Modeling Context Block. For more information, see The OpenSE Cookbook.

---

[22] Object Management Group (OMG). (2024). “System Modeling Language (SysML), Version 1.7.” (https://sysml.org/sysml-specs/)
[23] Karban, R.; Crawford, A.G.; Trancho, G.; Zamparelli, M.; Herzig, S.; Gomes, I.; Piette, M.; Brower, E. (2018). "The OpenSE Cookbook: A Practical, Recipe Based Collection of Patterns, Procedures, and Best Practices for Executable Systems Engineering for the Thirty Meter Telescope.” (https://trs.jpl.nasa.gov/handle/2014/48358)

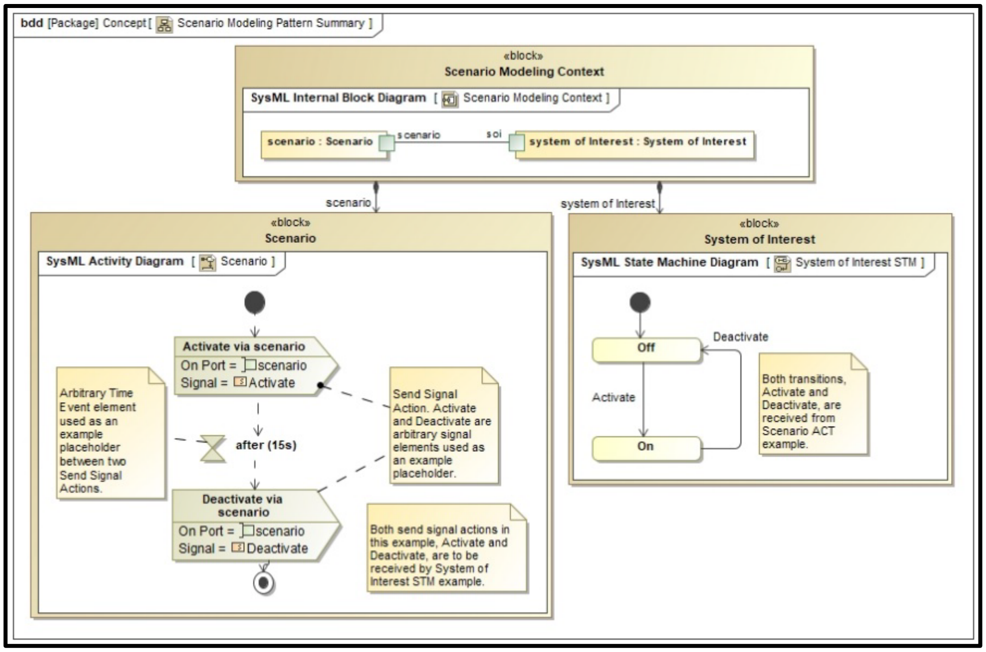


Figure D.3-1 Scenario Modeling Pattern Structure

D.4 System Specification Modeling

The System Specification modeling approach details a modeling method to relate elements in the Structure Pillar to the Requirements Pillar. In the system specification pattern, block additions for Logical Design, Logical Node Design, and Physical Design are added and trace to Systems, Subsystems, and Components (in the metamodel in section 7, **Figure 7-1**, this refers to any of the elements in the Structure Pillar) via a directed composition relationship. Another addition is the System Specification Block. This block is used to relate the structure blocks (Logical Design, Logical Node Design, and Physical Design) to the requirements. The System Specification Block uses a generalization to the structure elements and a directed composition to the requirement elements (that use a PBR requirement). For additional information, reference The OpenSE Cookbook. See the System Specification Block in **Figure D.5-1, Example Block Definition Diagram (bdd) of Another Modeling Approach for Requirements, Scenario, System Specification, and Verification,** for details on the relationships between requirements and the structure elements.

D.5 Verification Modeling

The OpenSE Cookbook details a requirements verification pattern. This Requirement Verification Pattern is structured to provide a platform to aid in Verification and Validation simulation. This pattern uses a Verification Context Block to relate the System Context element (similar to the one depicted in the metamodel in section 7, **Figure 7-1**) and a parametric diagram owned by the Verification Context Block. The System Context element has a part property that is used to define the scope of the verification analysis. The scope can include the System Specification Block described in the previous Section, the Scenario Modeling Context Block described earlier, or any other structure pillar element shown in **Figure 7-1**. See The OpenSE Cookbook for additional information. **Figure D.5-1** shows the Verification Context as it relates to the System Context.

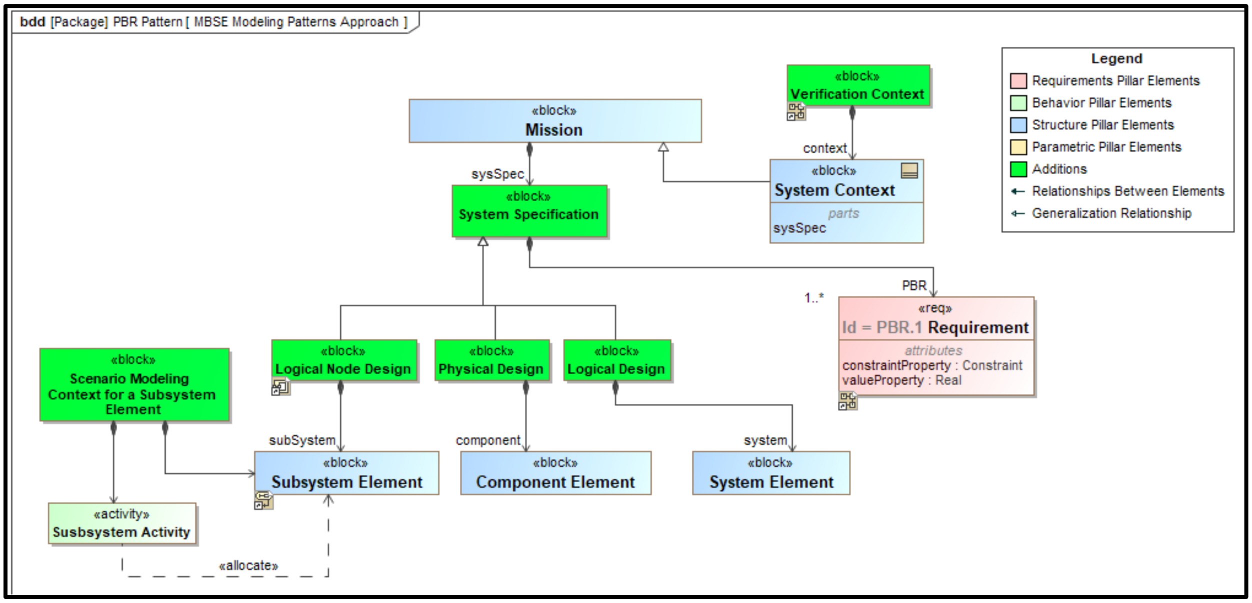


Figure D.5-1 Example Block Definition Diagram (bdd) of Another Modeling Approach for Requirements, Scenario, System Specification, and Verification

APPENDIX E: INTERFACE METAMODEL

E.1 Purpose

This appendix provides a metamodel of functional and structural interfaces.

E.2 Metamodel of Functional and Structural Interfaces

A metamodel is a depiction of the system modeling elements and their relationships. Section 7, **Figure 7-1** and **Figure 7-2**, shows the metamodel for system modeling based on NASA SE elements and relationships described in NPR 7123.1. **Figure E.2-1, Metamodel of Functional and Structural Interfaces**, shows how function elements interface with functions and how structural elements interface with other structural elements. It also shows the relationship to interface requirements.

In the metamodel, [ ] are used to capture the SysML® language-specific element or relationship type (block, activity, etc.).

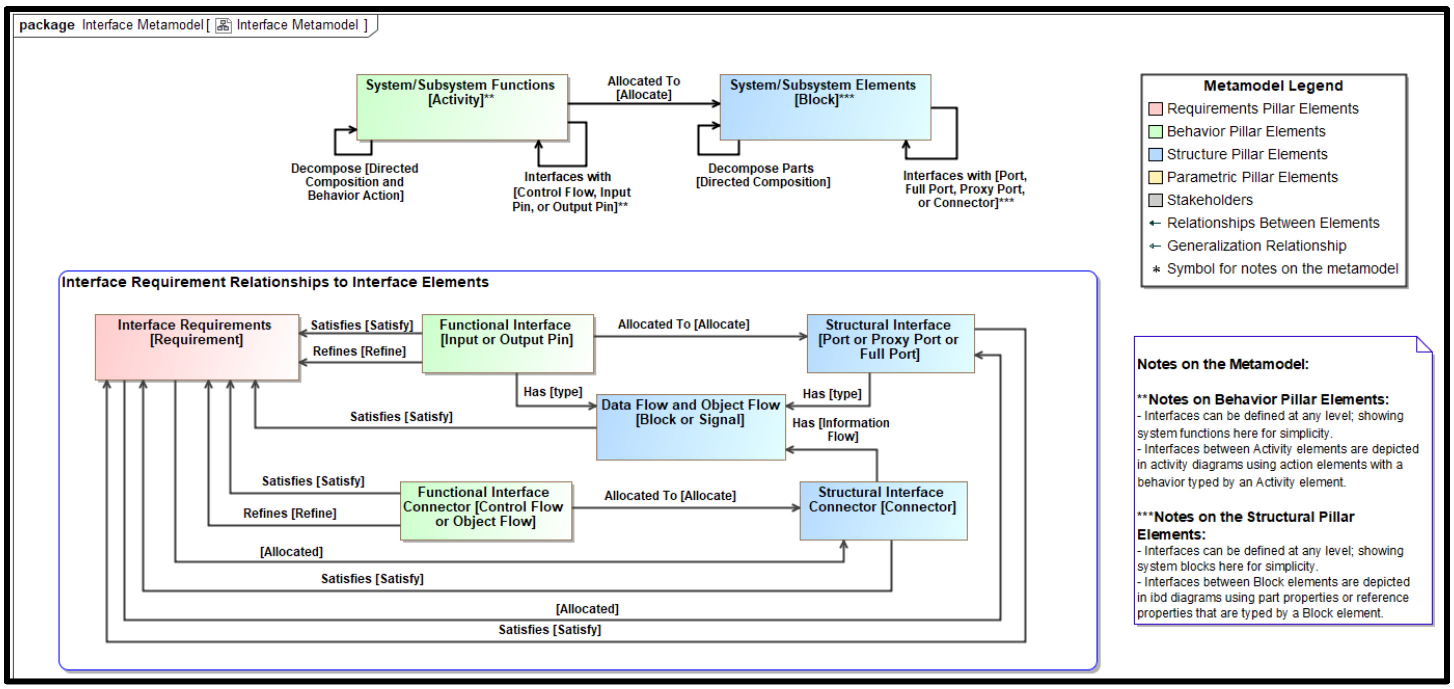


Figure E.2-1 Metamodel of Functional and Structural Interfaces

APPENDIX F: CONOPS TEMPLATE WITH MODEL CONTENT

F.1 Purpose

This appendix provides the Concept of Operations (ConOps) Annotated Outline from appendix S of the NASA Systems Engineering Handbook with example MBSE diagrams and tables to support each Section.

F.2 Concept of Operations Annotated Outline

The following ConOps outline and description is a replica from appendix S of the NASA Systems Engineering Handbook. The outline is provided in blue bold text and the descriptions are provided in italics. MBSE diagrams and tables are provided in sections where applicable/available and any additional commentary related to MBSE is provided in callout boxes. The diagrams and tables are pulled from sections 8 and 9 of this Handbook to bridge the modeling directly to a ConOps template.

Cover Page

Table of Contents

1.0 Introduction

1.1 Project Description

This section will provide a brief overview of the development activity and system context as delineated in the following two subsections.

1.1.1 Background

Summarize the conditions that created the need for the new system. Provide the high level mission goals and objective of the system operation. Provide the rationale for the development of the system.

1.1.2 Assumptions and Constraints

State the basic assumptions and constraints in the development of the concept. For example, that some technology will be matured enough by the time the system is ready to be fielded, or that the system has to be provided by a certain date in order to accomplish the mission.

1.2 Overview of the Envisioned System

This section provides an executive summary overview of the envisioned system. A more detailed description will be provided in Section 3.0

1009 Handbook Commentary for ConOps Section 1.2.1 Overview

Diagrams like a bdd or ibd can be used to describe overviews and system scope. A bdd can allow for an overview of system hierarchy and interface descriptions (via ports). An ibd can be used to depict an overview of element interactions (via ports, connectors, and item flows).

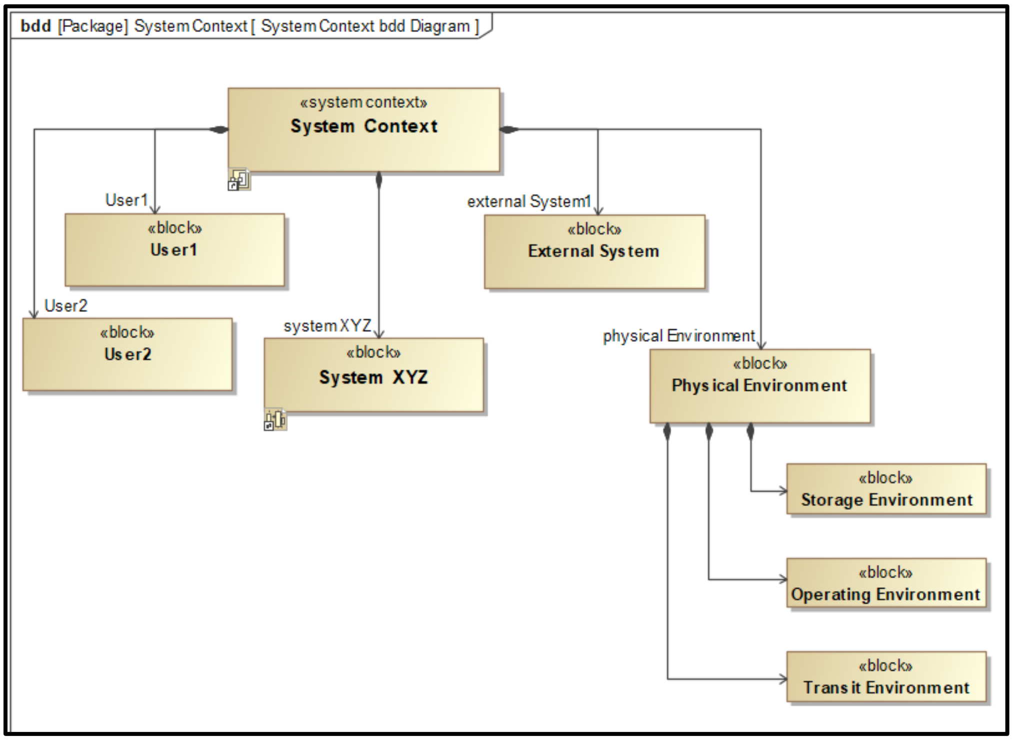


Figure F.2-1 System Context bdd Diagram (*Shown in section 8.3)*

Note: System Context can depict any level of structure to support the project/program. For example, the System Context can depict a mission, instrument level, subsystem level, or payload level.

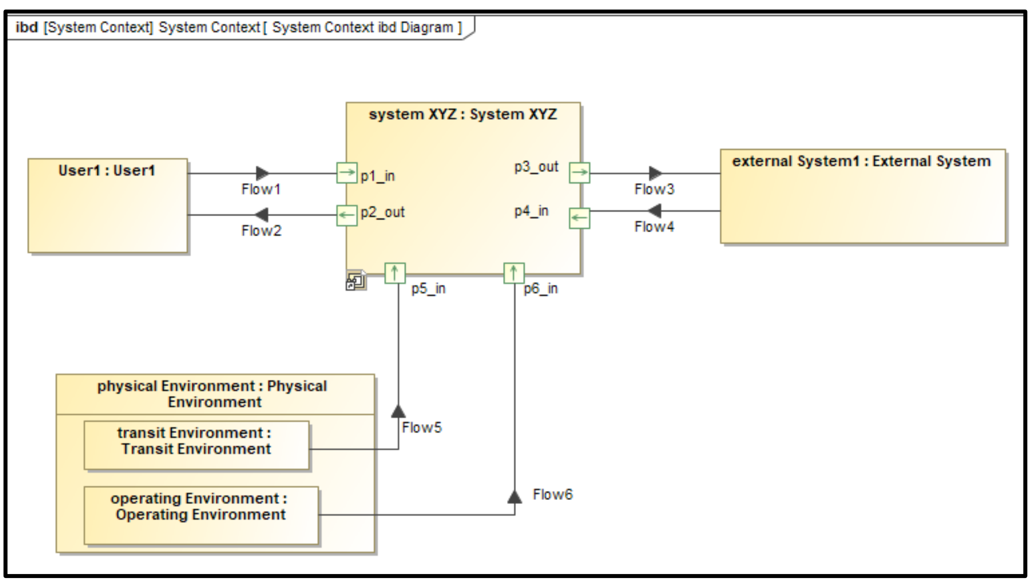


Figure F.2-2 System Context ibd Diagram (*Shown in section 8.4)*

1.2.1 Overview

This subsection provides a high-level overview of the system and its operation. Pictorials, graphics, videos, models, or other means may be used to provide this basic understanding of the concept.

1.2.2 System Scope

This section gives an estimate of the size and complexity of the system. It defines the system's external interfaces and enabling systems. It describes what the project will encompass and what will lie outside of the project's development.

2.0 Documents

2.1 Applicable Documents

This section lists the NASA/Government and non-NASA/Government specifications, standards, guidelines, handbooks, or other special publications applicable to this document.

2.2 Reference Documents

Reference documents are those documents that, though not a part of this guide, serve to clarify the intent and contents of this guide.

3.0 Description of Envisioned System

This section provides a more detailed description of the envisioned system and its operation as contained in the following subsections.

3.1 Needs, Goals and Objectives of Envisioned System

This section describes the needs, goals, and objectives as expectations for the system capabilities, behavior, and operations. It may also point to a separate document or model that contains the current up-to-date agreed-to expectations.

Id	Name	Text
need-1	R need-1 Need	Need to...
goal-1	R goal-1 Goal 1.1	Want to....
goal-2	R goal-2 Goal 1.2	Want to...
obj-1	R obj-1 Objective 1	Measure...
obj-2	R obj-2 Objective 2	Measure...
obj-3	R obj-3 Objective 3	Measure...

Figure F.2-3 Requirements Table of NGOs

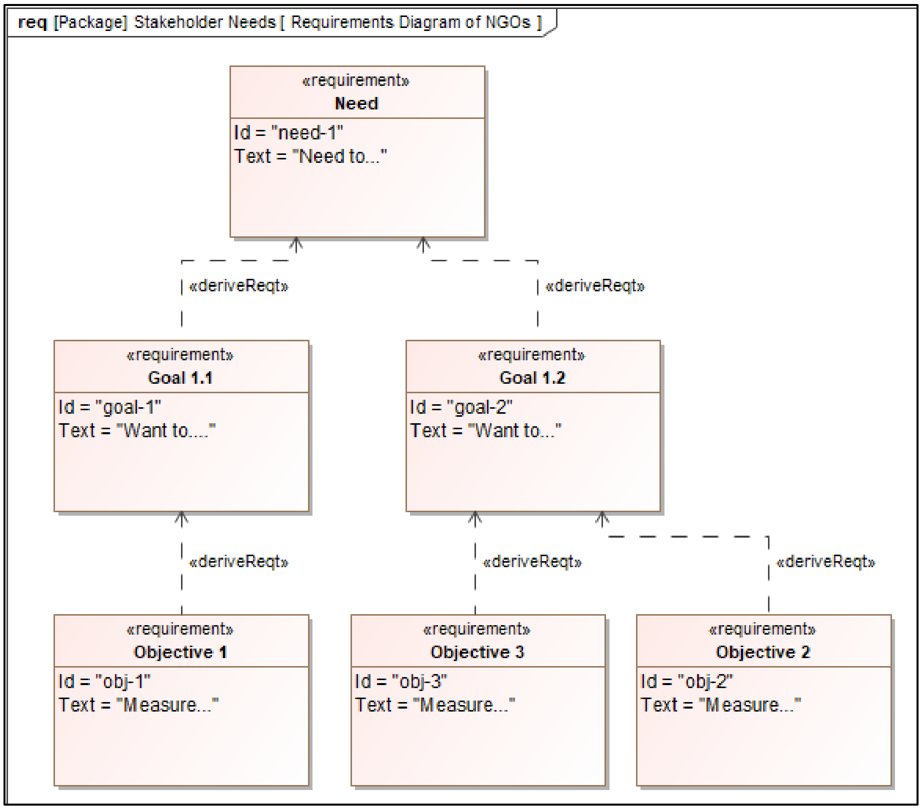


Figure F.2-4 Requirements Diagram of NGOs (*Shown in Section 8.2)*

3.2 Overview of System and Key Elements

This section describes at a functional level the various elements that will make up the system, including the users and operators. These descriptions should be implementation free; that is, not specific to any implementation or design but rather a general description of what the system and its elements will be expected to do. Graphics, pictorials, videos, and models may be used to aid this description.

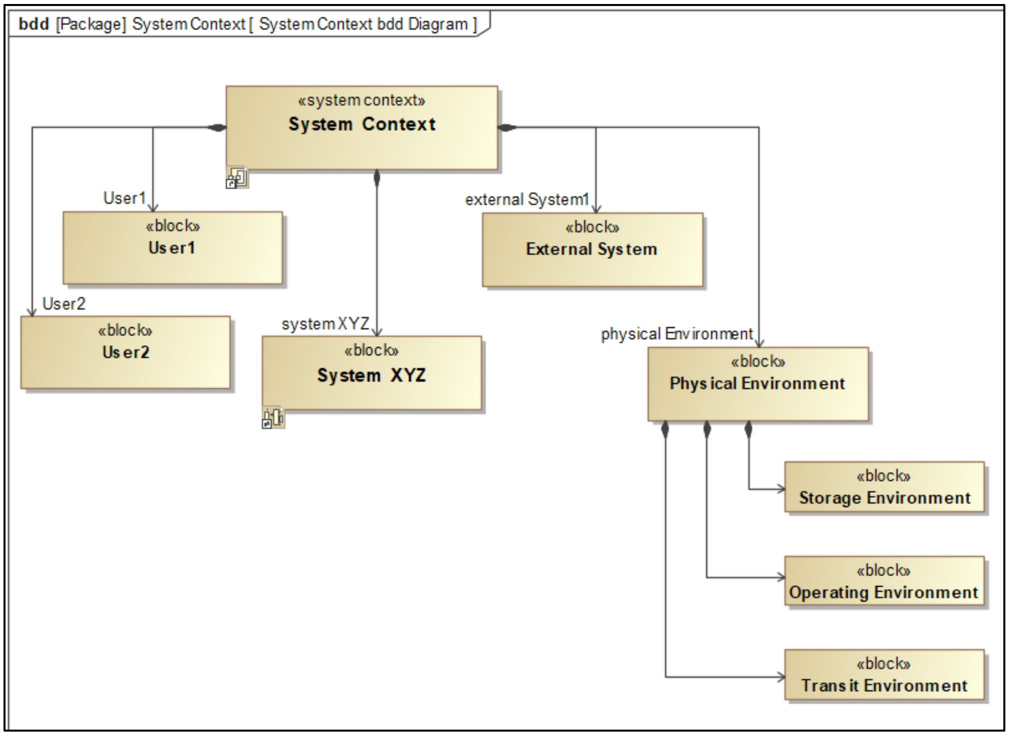


Figure F.2-5 System Context bdd Diagram (*Shown in section 8.3)*

Note: You can reference a diagram from ConOps section 1.2 rather than repeating the same image in this section.

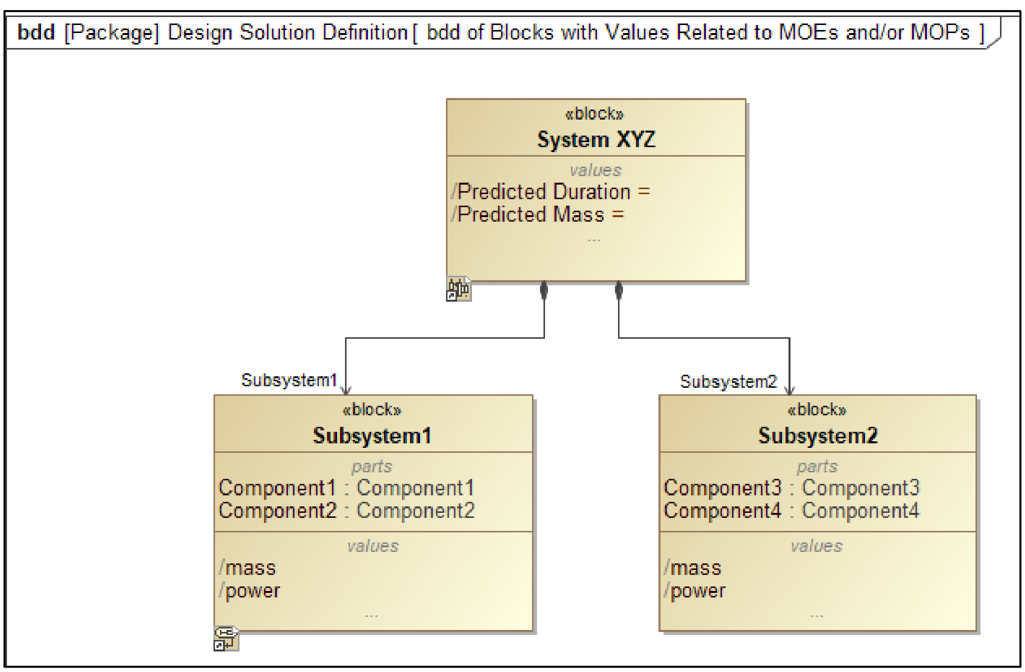


Figure F.2-6 bdd of System and Subsystem Blocks with Values Related to MOEs and MOPs

#	Name	Documentation	Allocated From
1	System XYZ	Description of System XYZ block	System Start Up(context System XYZ) System Shut Down(context System XYZ) System Function 1(context System XYZ) System Function 2(context System XYZ)
2	Subsystem1	Description of Subsystem1	Subsystem Function 1(context Subsystem1) Subsytem Function 3(context Subsystem1)
3	Subsystem2	Description of Subsystem2	Subsystem Function 2(context Subsystem2) Subsystem Function 4(context Subsystem2)

Figure F.2-7 Table Elements with Descriptions and Allocated Activities

A table can be used to depict the System and Key Elements along with a description of each (shown in the Documentation column) and the activities or functions that each element performs (shown in the Allocated From column).

#	Name	Associated Use Case	Allocated To	Associated Activity
1	User1	Execute Mission Behavior 1 Execute Mission Behavior 2 Execute Mission Behavior 3	User1	User Activity 1(context User1) User Activity 2(context User1)
2	User2	Execute Mission Behavior 3		

Figure F.2-8 Table of Actors traced to Activities and Use Cases

A table can be used to depict key users of the system. This table depicts information about the users of the system and what activities they perform and in what use cases they are involved in.

Note: This same table is shown in ConOps section 3.5 Proposed Capabilities as the Use Cases and Activities can be used to describe the capabilities of a system.

3.3 Interfaces

This section describes the interfaces of the system with any other systems that are external to the project. It may also include high-level interfaces between the major envisioned elements of the system. Interfaces may include mechanical, electrical, human user/operator, fluid, radio frequency, data, or other types of interactions.

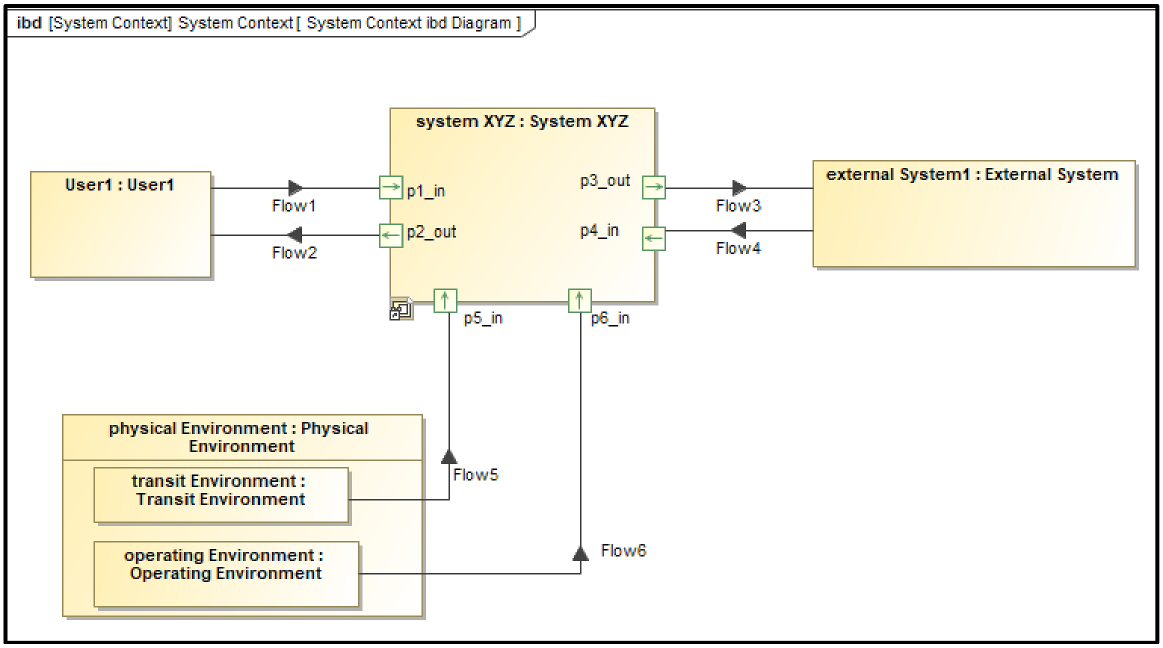


Figure F.2-9 System Context ibd Diagram (*Shown in Section 8.4)*

#	Part A	Port A	Item Flow	Direction	Part B	Port B	Documentation
1	system XYZ : ...	p1_in : ...	◄ Flow1	in	User1 : Model Organiz...		Interface additional details
2	system XYZ : ...	p2_out : .	► Flow2	out	User1 : Model Organiz...		
3	system XYZ : ...	p3_out : .	► Flow3	out	external System1 : Mo...		
4	system XYZ : ...	p4_in : ...	◄ Flow4	in	external System1 : Mo...		
5	system XYZ : ...	p5_in : ...	◄ Flow5	in	physical Environment : ..		
6	system XYZ : ...	p6_in : ...	◄ Flow6	in	physical Environment : ..		

Figure F.2-10 Table of System Context Interfaces

#	Name	Documentation
1	Flow1	Description of Flow1...
2	Flow2	Description of Flow2...
3	Flow3	Description of Flow3...
4	Flow4	Description of Flow4...
5	Flow5	Description of Flow5...
6	Flow6	Description of Flow6...

Figure F.2-11 Table of Interface Items and Descriptions

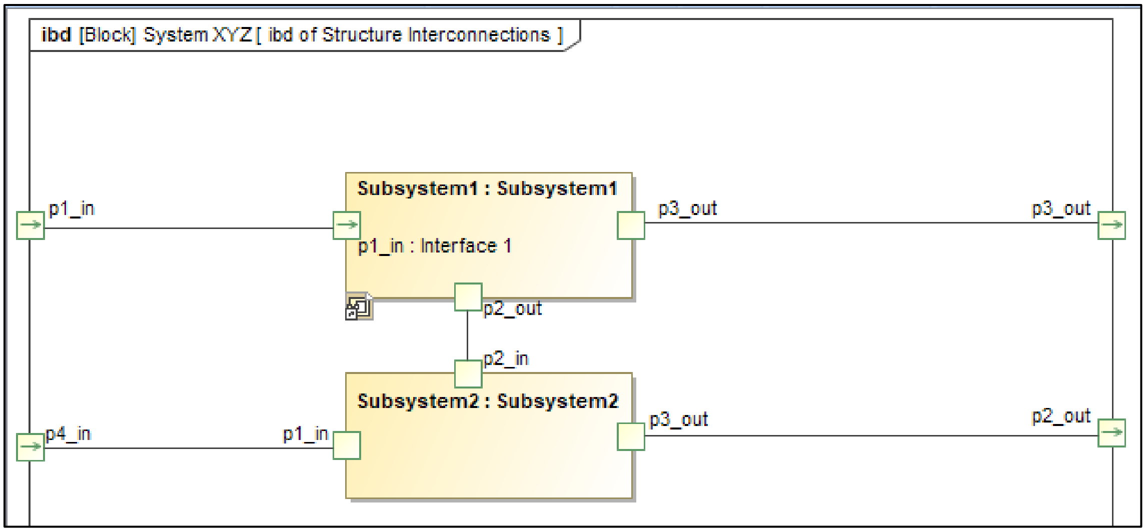


Figure F.2-12 ibd of Structure Interconnections (*Shown in section 8.8)*

> Note: The above picture shows a few different port depictions for ibds. The modeler can display port types or omit them for visual simplicity depending on how the diagram is being used. Details about the port types can be captured in supporting tables as well.

3.4 Modes of Operations

This section describes the various modes or configurations that the system may need in order to accomplish its intended purpose throughout its life cycle. This may include modes needed in the development of the system, such as for testing or training, as well as various modes that will be needed during it operational and disposal phases.

Modeling for Modes of Operations

Types of diagrams and tables that can be helpful in this section include:

- State Machines that show the system's modes and how the system transitions from one mode to another; these diagrams can also capture the behavior that occurs in and between the modes.
- Activity Diagrams or Sequence Diagrams that show each system involved in each mode (via swim lanes and lifelines).
- Table of modes, mode description, functions occurring in each mode, conditions for each mode, etc.
- Matrices that show traceability between a mode and:
  - Mission phases (ex: allocating a mode (State) to a mission phase element (Activity).
  - Scenarios/ use cases (ex: allocating a scenario/ use case (Activity or Use Case Element for example) to a mode (State)).

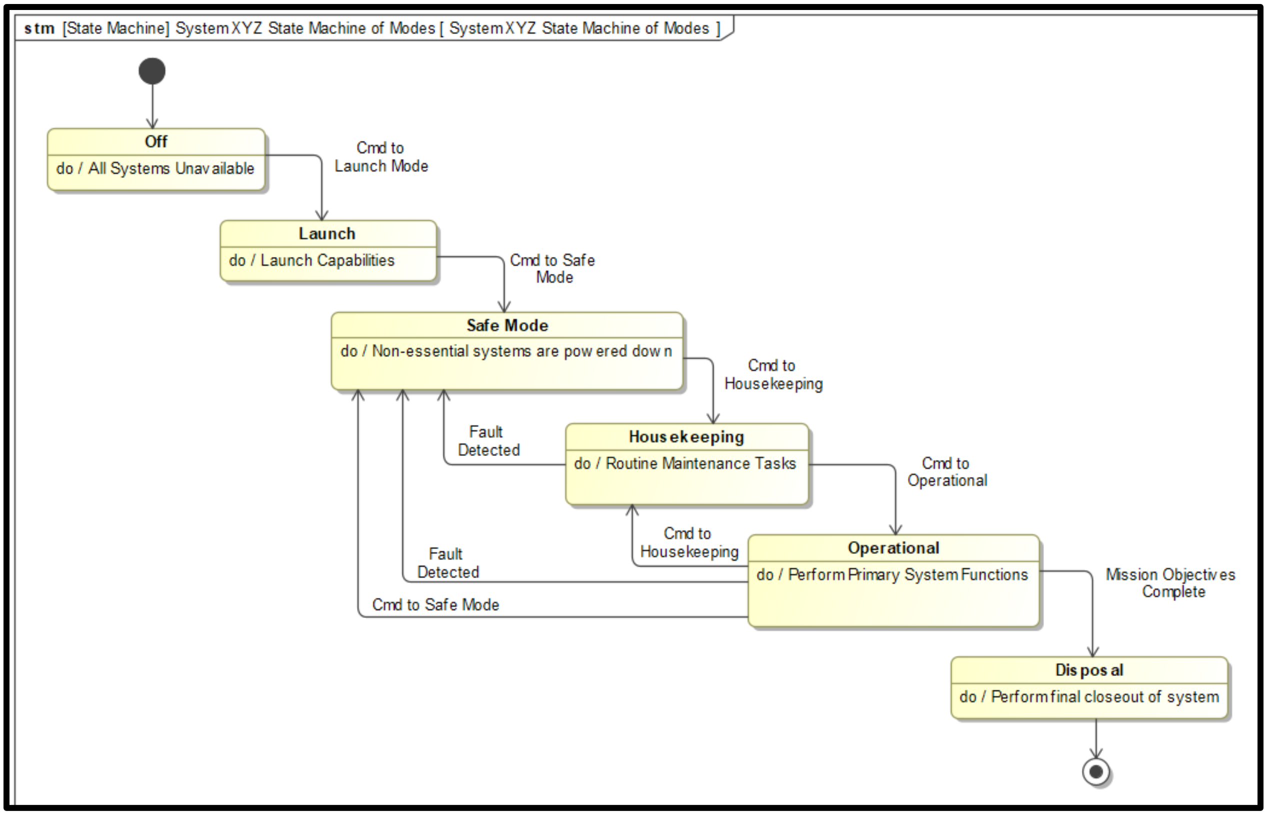


Figure F.2-13 System XYZ State Machine of Modes

Notes: In this example only "do" activities are demonstrated on the example mode state and the transitions are demonstrated with Signal events. These diagrams can leverage much more of the State Machine depictions (ex: entry and exit details or guards on transitions).

The "do" activities in this example are using Activity elements that can be used to describe flow of events and systems involved in each mode.

#	Name	Description	Transition Conditions	Function Description During Mode
1	Off	All systems in off state and unavailable.		All Systems Unavailable
2	Launch	Configuration during ascent to destination. Only functionality available during this mode would be functions required for survival during launch phase.	Cmd to Launch Mode	Launch Capabilities
3	Housekeeping	Mode to support routine maintenance activities such as checkout, orbit adjustments, sensor calibrations.	Cmd to Housekeeping	Routine Maintenance Tasks
4	Operational	Performing primary mission objectives and systems operating nominally.	Cmd to Operational	Perform Primary System Functions
5	Safe Mode	A protective state triggered by anomalous condition or commanded to for a controlled mode change.	Fault Detected Cmd to Safe Mode	Non-essential systems are powered down
6	Disposal	Final closeout of the system when the mission has been completed.	Mission Objectives Complete	Perform final closeout of system

Figure F.2-14 Modes and Mode Descriptions

Notes:

- A table like that shown in Figure F.2-14 can be used to describe the mode, key functions or systems available during that mode, and the transitions conditions (criteria or events that trigger a transition to a mode).
- Mode functions can be captured via multiple behavior types (activities, interactions, state machines, etc.) in the mode as an entry, exit or do activity.
- In this state machine example transitions are using signal events; however, other event types can be specified as needed. Transitions can also depict guards and behaviors.

3.5 Proposed Capabilities

This section describes the various capabilities that the envisioned system will provide. These capabilities cover the entire life cycle of the system's operation, including special capabilities needed for the verification/validation of the system, its capabilities during its intended operations, and any special capabilities needed during the decommissioning or disposal process.

1009 Handbook Commentary for ConOps Section 3.5 Proposed Capabilities

This section details what the system needs to meet the objectives. Modeling in this section gets into more details about the behavior of the system. Diagram examples that can be leveraged to assist with capability descriptions include Use Cases, Functional/Activity decomposition diagrams, and tables describing the capabilities.

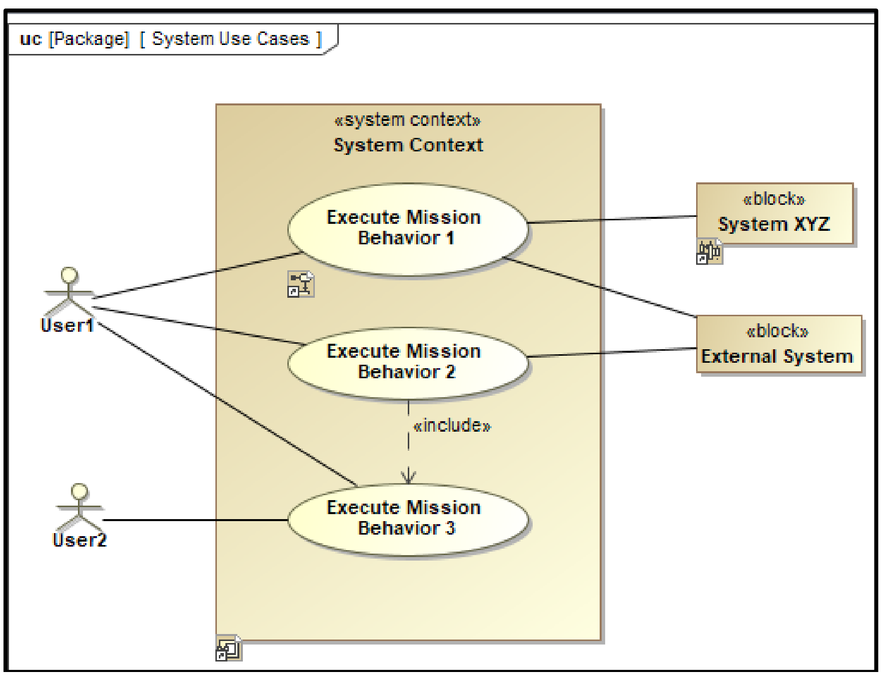


Figure F.2-15 System Use Cases (*Shown in section 8.5)*

Notes on Use Case Diagrams:

- Actors can represent roles played by human users, external hardware, or any other subjects.
- The <<include>> relation is used to indicate where one Use Case may be a subset of another.

Example content for the above diagram:

- An example of the actor "User1" depicted in this Use Case diagram could be an automation system.
- An example of the Use Case "Execute Mission Behavior 1" could be "Fly a desired trajectory."
- An example of "System XYZ" could be the spacecraft or aircraft of interest.

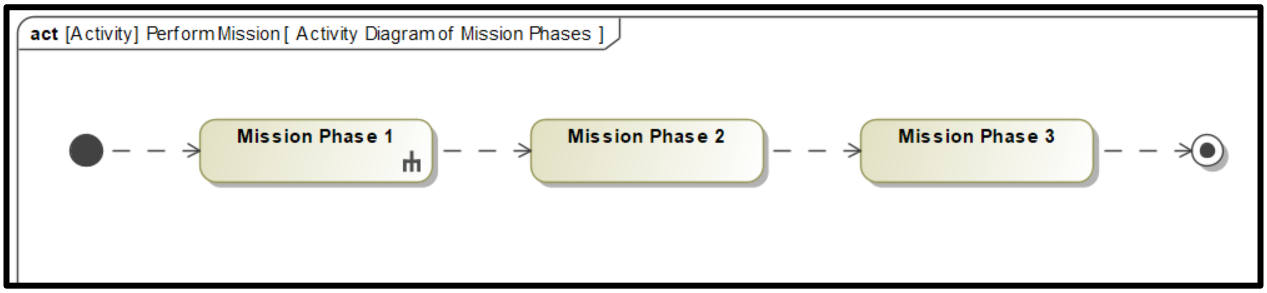


Figure F.2-16 Activity Diagram of Mission Phases

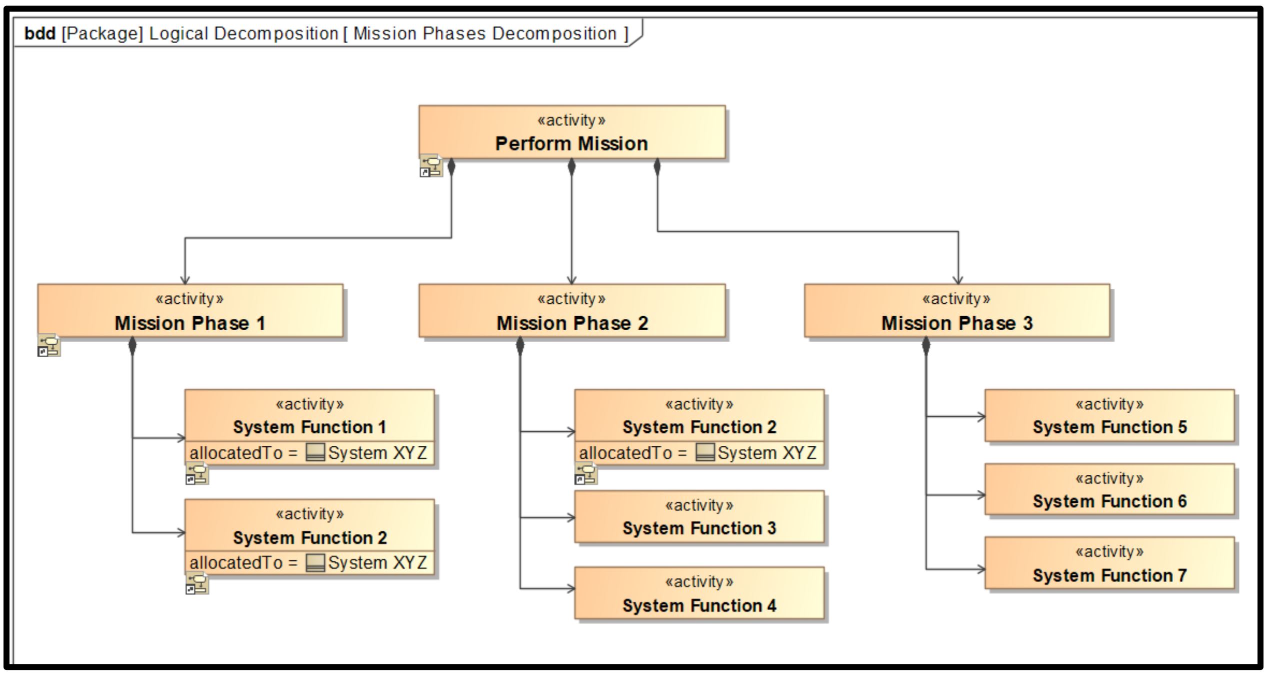


Figure F.2-17 bdd of Mission Phase and Activity Decomposition

Notes:

- The top level activity pertains to the scope of the domain for a given project.
- Mission Phases in this example are shown using Activity elements; however, depending on the needs of the project model, behavior can also be represented with State Machines and/or Sequence Diagrams.

- State Machines are helpful when describing states, state transitions and triggers (more than the flow of events/activities). However, the modeler can also depict the behavior occurring in a given state.
- The Sequence Diagram is useful for depicting the interactions between system components and the order of the interactions.

#	Name	Associated Use Case	Allocated To	Associated Activity
1	User1	Execute Mission Behavior 1 Execute Mission Behavior 2 Execute Mission Behavior 3	User1	User Activity 1(context User1) User Activity 2(context User1)
2	User2	Execute Mission Behavior 3		

Figure F.2-18 Table of Actors traced to Activities and Use Cases

This example table shows Use Cases and Activities associated with the user of the system. Use Cases and Activities can be used to describe the capabilities of a system.

Note: The same table is shown in ConOps section 3.2 Overview of System and Key Elements as key users and what their roles are can be described with a similar type of table.

#	Name	Allocated To	Documentation
1	User Activity 1	User1	
2	User Activity 2	User1	
3	System Function 1	System XYZ	
4	System Function 2	System XYZ	
5	Subsystem Function 1	Subsystem1	
6	Subsystem Function 2	Subsystem2	

Figure F.2-19 Table of Activities with Allocated Elements

4.0 Physical Environment

This section should describe the environment that the system will be expected to perform in throughout its life cycle, including integration, tests, and transportation. This may include expected and off-nominal temperatures, pressures, radiation, winds, and other atmospheric, space, or aquatic conditions. A description of whether the system needs to operate, tolerate with degraded performance, or just survive in these conditions should be noted.

1009 Handbook Commentary for ConOps Section 4.0 Physical Environment

The intention for modeling in support of Section 4.0 Physical Environment is to capture model content that will drive your design; it does not need to exhaustively document all physical environments for which the system interacts.

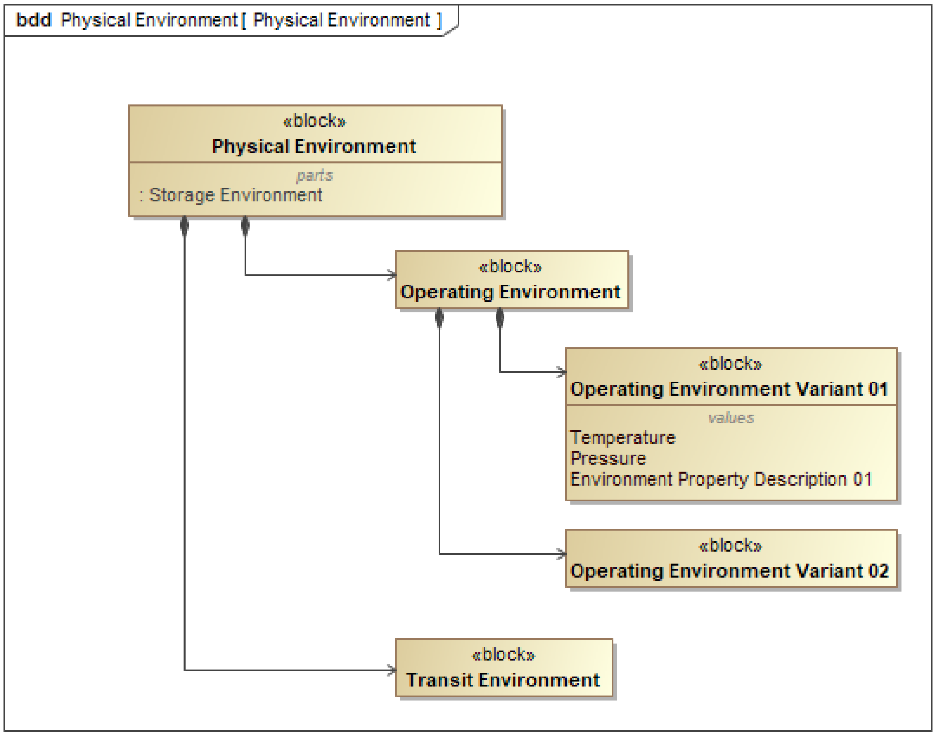


Figure F.2-20 Physical Environment bdd *(Subset of diagram shown in section 8.3)*

5.0 Support Environment

This section describes how the envisioned system will be supported after being fielded. This includes how operational planning will be performed and how commanding or other uploads will be determined and provided, as required. Discussions may include how the envisioned system would be maintained, repaired, replaced, it's sparing philosophy, and how future upgrades may be performed. It may also include assumptions on the level of continued support from the design teams.

1009 Handbook Commentary for ConOps Section 5.0 Support Environment

Modeling for section 5.0 Support Environment can be similar to modeling shown in section 4.0 Physical Environment. Other diagrams that can be included in this Section include Activity Diagrams to depict system maintenance and update activities. As for those activity diagrams, whether they should be included in this section or in the main body of section 6.0, Operational Scenarios, Use Cases and/or Design Reference Mission, will depend on if these systems are part of the system of interest, or supporting infrastructure leveraged by the system of interest. Supporting infrastructure could be depicted here.

6.0 Operational Scenarios, Use Cases and/or Design Reference Missions

This section takes key scenarios, use cases, or DRM and discusses what the envisioned system provides or how it functions throughout that single-thread timeline. The number of scenarios, use cases, or DRMs discussed should cover both nominal and off-nominal conditions and cover *all expected functions and capabilities. A good practice is to label each of these scenarios to facilitate requirements traceability; e.g., [DRM-0100], [DRM-0200].*

1009 Handbook Commentary for ConOps Section 6.0

Modeling for this section can include behavior diagrams such as Use Case Diagrams, Activity Diagrams, State Machine Diagrams, as well as Sequence Diagrams.

- Scenarios can be captured via use case, activity, or interaction elements.
- The sequence of events for each scenario can be captured in Activity, Sequence, or State Machine Diagrams depending on the needs of the project model.

Note: Section 6.0 could be described using the next level diagram decomposition from those used in section 3.0. For example, section 3.0 may have had mission phases, and this section would show the scenarios and uses cases within each of those phases.

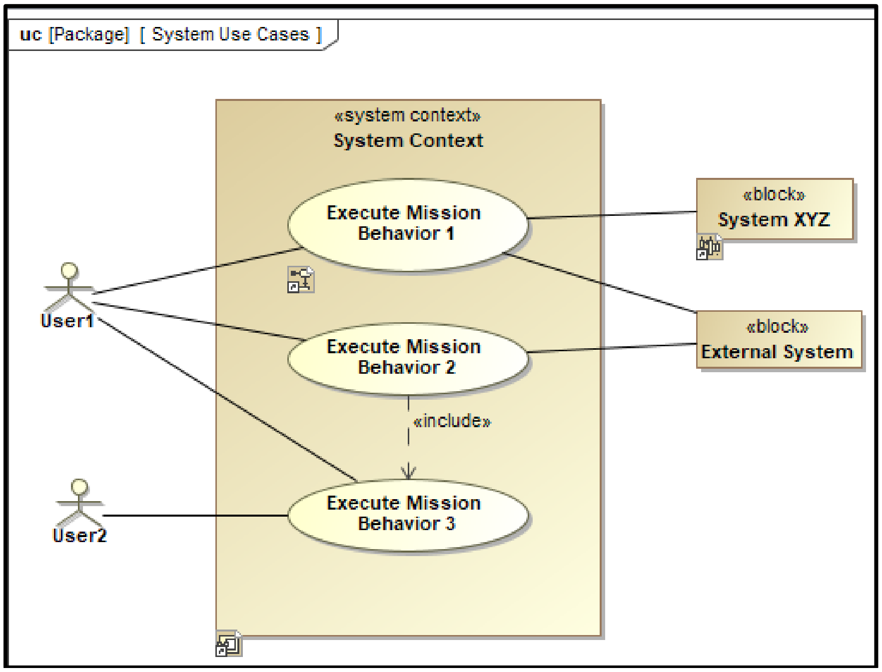


Figure F.2-21 System Use Cases (*Shown in Section 8.5)*

Notes:

- *This same Use Case diagram was shown in ConOps section 3.5 Proposed Capabilities as an example of use cases depicting capabilities; in this section, each of these Use Cases can be decomposed to lower level use cases to capture all the functions and capabilities (both nominal and off-nominal).*
- Activity Diagrams or Sequence Diagrams can be used to describe the steps/ flow of events for each use case; An activity diagram for Execute Mission Behavior 1 is shown in Figure F.2-22.
- Activity elements can be decomposed as well to lower level capabilities and functions independent of use cases; this can be decided by preference of the modeler or stakeholders

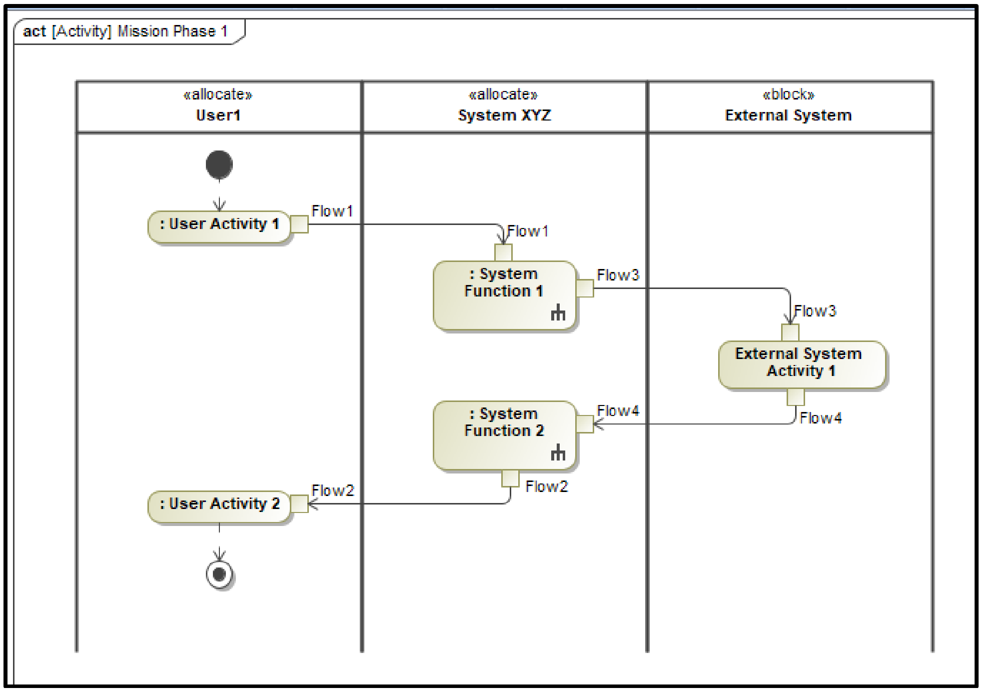


Figure F.2-22 Activity Diagrams to Support ConOps Use Cases (*Shown in section 8.6)*

In this example, the activity diagram further explains the details of the use case example "Execute Mission Behavior 1". Activities can be captured in activity diagrams showing interactions between activities and allocations to structure elements.

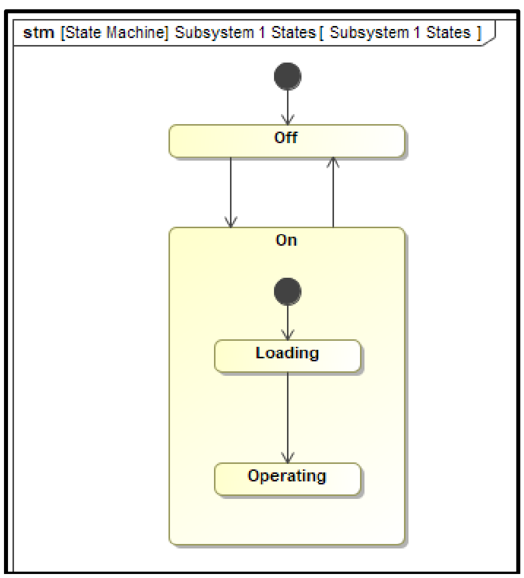


Figure F.2-23 Subsystem 1 States

State Machines in this section can be utilized to describe states/modes of the systems and what scenarios or capabilities are provided in the given state. The transitions can describe what scenario or activity triggers a state change.

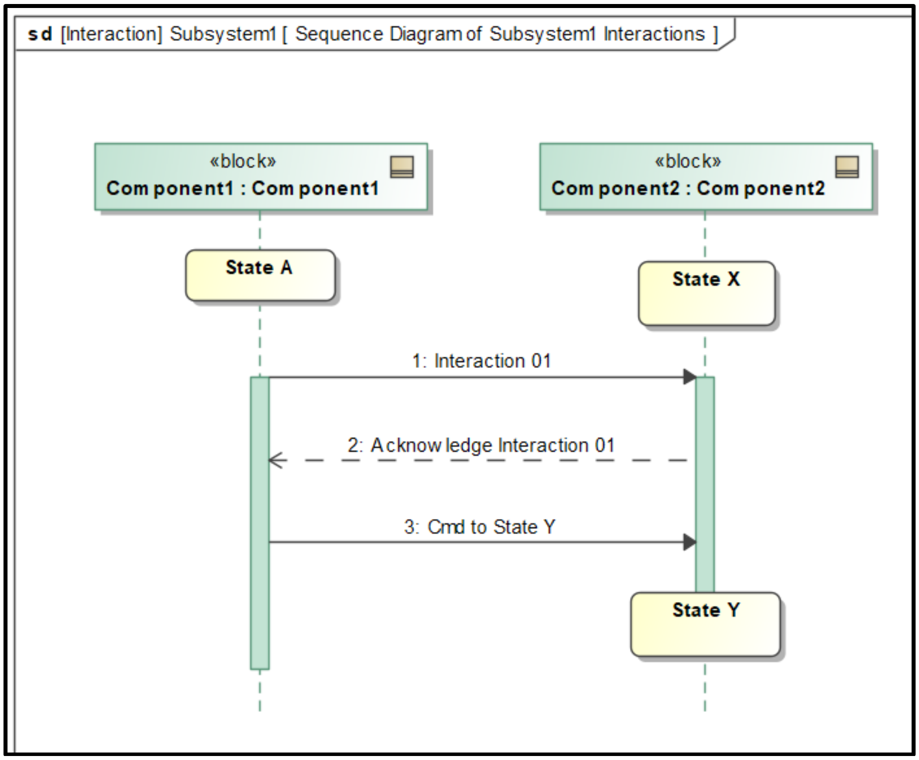


Figure F.2-24 Sequence Diagram Example

> This is an example sequence diagram showing interactions between two components of Subsystem 1. States are shown on a lifeline to provide more descriptive information about the interactions between elements and what states those elements are in when those interactions are occurring.

6.1 Nominal Conditions

These scenarios, use cases, or DRMs cover how the envisioned system will operate under normal circumstances where there are no problems or anomalies taking place.

6.2 Off-Nominal Conditions

These scenarios cover cases where some condition has occurred that will need the system to perform in a way that is different from normal. This would cover failures, low performance, unexpected environmental conditions, or operator errors. These scenarios should reveal any additional capabilities or safeguards that are needed in the system.

> **1009 Handbook Commentary for ConOps Section 6.2**
>
> The same modeling methods used for modeling nominal conditions can be applied to off-nominal conditions.

7.0 Impact Considerations

This section describes the potential impacts, both positive and negative, on the environment and other areas.

7.1 Environmental Impacts

Describes how the envisioned system could impact the environment of the local area, state, country, worldwide, space, and other planetary bodies as appropriate for the systems intended purpose. This includes the possibility of the generation of any orbital debris, potential contamination of other planetary bodies or atmosphere, and generation of hazardous wastes that will need disposal on earth and other factors. Impacts should cover the entire life cycle of the system from development through disposal.

7.2 Organizational Impacts

Describes how the envisioned system could impact existing or future organizational aspects. This would include the need for hiring specialists or operators, specialized or widespread training or retraining, and use of multiple organizations.

7.3 Scientific/Technical Impacts

This subsection describes the anticipated scientific or technical impact of a successful mission or deployment, what scientific questions will be answered, what knowledge gaps will be filled, and what services will be provided. If the purpose of this system is to improve operations or logistics instead of science, describe the anticipated impact of the system in those terms.

8.0 Risks and Potential Issues

This section describes any risks and potential issues associated with the development, operations or disposal of the envisioned system. Also includes concerns/risks with the project schedule, staffing support, or implementation approach. Allocate subsections as needed for each risk or issue consideration. Pay special attention to closeout issues at the end of the project.

1009 Handbook Commentary for ConOps Section 8.0

It can be useful to associate a risk or a potential issue element to any part of the system model that they impact. This can be done using a block element with value properties to represent impact or likelihood, etc. This can then be reported in tabular form for section 8.0, with the risk and its associated diagrams or design elements. There are many modeling methods for this not included in this version of the handbook.

Appendix A Acronyms

This part lists each acronym used in the ConOps and spells it out.

Note: Acronyms and their definition can be captured in a table in the model.

Appendix B Glossary of Terms

The part lists key terms used in the ConOps and provides a description of their meaning.

Note: Glossary of Terms and their definition can be captured in a table in the model.

APPENDIX G: ADDITIONAL MODEL-BASED DOCUMENTS

G.1 Purpose

This appendix describes additional NASA Standards, Handbooks, and repositories that capture model-based context along with a brief description of each.

G.2 NASA Documents With Model-Based Context

Table G.2-1 NASA Standards, Handbooks, and Repositories with Model-Based Context

Document Number	Title	Description
NASA-HDBK-1004	NASA Digital Engineering Acquisition Framework Handbook	Provides guidance for establishing NASA's digital engineering acquisition framework that includes Data Requirements Descriptions (DRDs) and contractual language for the Statement of Work (SOW) in support of a digital engineering environment (DEE).[24]
NASA-HDBK-1005	NASA Space Mission Architecture Framework (SMAF) Handbook For Uncrewed Space Missions	Describes the rules of which an architecture should be described to satisfy stakeholders.[24]
NASA-HDBK-7009	NASA Handbook For Models and Simulations: An Implementation Guide For NASA-STD-7009	A companion guide to NASA-STD-7009A. Provides technical information, clarification, examples, processes, and techniques to help institute good modeling and simulation practices in NASA.[24]
NASA-STD-7009	Standards for Models and Simulations	Establishes uniform practices in modeling and simulation to ensure essential requirements are applied to their design, development, and use while ensuring acceptance criteria are defined by the program/project and approved by the delegated NASA Technical Authority.[24]
N/A	Model-Based System Engineering, NEN (https://nen.nasa.gov/web/mbse)	NASA MBSE repository with shared MBSE resources.

[24] NASA Technical Standards; https://standards.nasa.gov/NASA-Technical-Standards

APPENDIX H: ACRONYMS, ABBREVIATIONS, AND DEFINITIONS

H.1 Purpose

This appendix provides a list of applicable acronyms, abbreviations, and definitions as used in this Handbook.

H.2 Acronyms and Abbreviations

act	activity diagram
bdd	block definition diagram
ConOps	concept of operations
HDBK	Handbook
ibd	internal block diagram
IEC	International Electrotechnical Commission
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
MBSE	Model-Based Systems Engineering
MOE	measure of effectiveness
MOP	measure of performance
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
NEN	NASA Engineering Network
NGO	needs, goals, and objectives
NPR	NASA Procedural Requirements
OMG®	Object Management Group®
OOSEM	Object-Oriented Systems Engineering Method
par	parametrics diagram
PBR	property-based requirement(s)
PBS	product breakdown structure
pkg	package diagram
req	requirement diagram
sd	sequence diagram
SE	systems engineering
SEMP	Systems Engineering Management Plan
SP	Special Publication
STD	standard
stm	state machine
SysML®	Systems Modeling Language™
TPM	technical performance measures
uc	use case diagram
UML®	Unified Modeling Language
V&V	verification and validation

H.3 Definitions

Activity: A set of tasks that describe the technical effort to accomplish a process and help generate expected outcomes. (Source: NASA/SP-2016-6105, Revision 2)

Analysis:

a. In SE, use of mathematical modeling and analytical techniques to predict the compliance of a design to its requirements based on calculated data or data derived from lower system structure end-product validations. (Source: NASA/SP-2016-6105, Revision 2)
b. In the design process, the examination of a situation or problem to understand the item in question and make appropriate recommendations. (Source: NASA-HDBK-7009)

Artifact: Any product produced by the project team, e.g., requirements, documents, help systems, code, executables, test documentation, test results, records, and diagrams. (Source: NASA-STD-7009)

Behavior: The effect produced when an instance of a complex system or organism is used in its operational environment. (Source: SEBoK)

Concept of Operations (ConOps): Describes the overall high-level concept of how the system will be used to meet stakeholder expectations, usually in a time-sequenced manner. (Source: NASA/SP-2016-6105, Revision 2)

Constraint: A condition dictated by external factors such as orbital mechanics, an existing system that must be utilized (external interface), a regulatory restriction, state of technology, or result of the overall budget environment that is to be met. It typically cannot be changed based on trade-off analysis.

Design Solution Definition Process: A process that translates the outputs of the Logical Decomposition Process into a design solution definition that is in a form consistent with the product life-cycle phase and product layer location in the system structure and that will satisfy phase success criteria. (Sources: NPR 7123.1 and NASA/SP-2016-6105, Revision 2).

Logical Decomposition Process: A process used to improve understanding of the defined technical requirements and the relationships among the requirements (e.g., functional, behavioral, performance, and temporal) and to transform the defined set of technical requirements into a set of logical decomposition models and their associated set of derived technical requirements for lower levels of the system and for input to the Design Solution Process. (Sources: NPR 7123.1 and NASA/SP-2016-6105, Revision 2)

Measure of Effectiveness (MOE): A measure by which a stakeholder's expectations are judged in assessing satisfaction with products or systems produced and delivered in accordance with the associated technical effort, deemed critical to both acceptability of product by stakeholder and operational/mission usage, typically quantitative in nature or not able to be used directly as a design-to requirement. (Source: NPR 7123.1)

Measure of Performance (MOP): A quantitative measure that, when met by the design solution, will help ensure that an MOE for a product or system will be satisfied. MOPs are given special attention during design to ensure that the MOEs with which they are associated are met. There are generally two or more measures of performance for each MOE. (Source: NPR 7123.1)

Metamodel: A model of a model that describes the concepts in the modeling language, their characteristics, and interrelationships. (Source: Friedenthal, S.; Moore, A.; and Steiner, R. (2014). "A Practical Guide to SysML: The Systems Modeling Language," 3rd ed. Boston: Morgan Kaufmann.)

Model: A description or representation of a system, entity, phenomena, or process. (Note: A model may be constructed from multiple sub-models; the sub-models and the integrated sub-models are all considered models. Likewise, any data that go into a model are considered part of the model.) (Source: NASA-HDBK-7009)

Model-Based Systems Engineering (MBSE): The formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life-cycle phases. (Source: INCOSE - International Council on Systems Engineering. (n.d.). Retrieved October 4, 2022. "INCOSE Initiatives". INCOSE. (https://www.incose.org/incose-member-resources/initiatives)

Modeling:
a. The act of creating a system representation (i.e., the act of creating a model);
b. The act of utilizing a system representation (i.e., utilizing a model) as an approach for analyses. (Source: NASA-HDBK-7009)

Object Management Group® (OMG®): An international non-profit technology standards consortium that helped design modeling standards such as SysML®. (Source: OMG®, https://www.omg.org/about/index.htm)

Object-Oriented Systems Engineering Method (OOSEM): A systems-level development method that combines object-oriented concepts with traditional SE practices. (Source: INCOSE, https://www.incose.org/communities/working-groups-initiatives/object-oriented-se-method)

Pattern: A documented and structured scalable and reusable essence of good practice that seeks to address a problem or a group of problems.

Process: A set of activities used to convert inputs into desired outputs to generate expected outcomes and satisfy a purpose. (Sources: NPR 7123.1 and NASA/SP-2016-6105, Revision 2)

Program: A strategic investment by a Mission Directorate (or mission support office) that has defined goals, objectives, architecture, funding level, and a management structure that supports one or more projects. (Source: NPR 7123.1)

Project: A specific investment having defined goals, objectives, requirements, life-cycle cost, a beginning, and an end. (Sources: NPR 7123.1 and NASA/SP-2016-6105, Revision 2)

Requirement: The agreed-upon need, desire, want, capability, capacity, or demand for personnel, equipment, facilities, or other resources or services by specified quantities for specific periods of time or at a specified time expressed as a "shall" statement. (Sources: NPR 7123.1 and NASA/SP-2016-6105, Revision 2)

Scenario: The description or definition of the relevant system and environmental assumptions, conditions, or parameters used to derive the course of events during the analysis of a model. (Source: Modified from NASA-HDBK-7009)

Simulation: The imitation of the behavioral characteristics of a system, entity, phenomena, or process. (Source: NASA-HDBK-7009)

Specification: An element that prescribes completely, precisely, and verifiably the requirements, design, behavior, or characteristics of a system or system component, usually in the form of a requirement. (Source: Modified from NPR 7123.1)

Stakeholder: A group or individual who is affected by or has an interest or stake in a program or project. (Source: NPR 7123.1)

Stakeholder Expectations Definition Process: A process used to elicit and define use cases, scenarios, concept of operations (ConOps), and stakeholder expectations for the applicable product life-cycle phases and product later. (Sources: NPR 7123.1 and NASA/SP-2016-6105, Revision 2)

System: The combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (Sources: NPR 7123.1)

Systems Engineering (SE): NASA SE is a logical systems approach performed by multidisciplinary teams to engineer and integrate NASA's systems to ensure NASA products meet the customer's needs. Implementation of this systems approach will enhance NASA's core engineering capabilities while improving safety, mission success, and affordability. This systems approach is applied to all elements of a system (i.e., hardware, software, and human) and all hierarchical levels of a system over the complete program/project life cycle. (Source: NPR 7123.1)

Systems Engineering (SE) Engine: The SE model that provides the 17 technical processes defined in NPR 7123.1 and their relationships with each other. (Source: NPR 7123.1)

Systems Modeling Language™ (SysML®): A general-purpose modeling language developed by OMG® for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis methods. (Object Management Group (OMG). (2022). "What is SysML?" OMG SysML. (https://www.omgsysml.org/what-is-sysml.htm))

System of Interest: The system whose characteristics are under consideration regardless of where it lies in the product hierarchy. (Source: NPR 7123.1)

Tailoring: The process used to seek relief from SE NPR requirements consistent with program or project objectives, allowable risk, and constraints. (Source: NPR 7123.1)

Technical Performance Measures (TPM): A set of performance measures that are monitored by comparing the current actual achievement of the parameters with that anticipated at the current time and on future dates. (Source: NPR 7123.1)

Technical Requirements: The requirements that capture the characteristics, features, functions, and performance that the end product will have to meet stakeholder expectations. (Source: NPR 7123.1)

Technical Requirements Definition Process: A process used to transform the stakeholder expectations into a complete set of validated technical requirements expressed as "shall" statements that can be used for defining a design solution for the product breakdown structure (PBS) model and related enabling products. (Sources: NPR 7123.1 and NASA/SP-2016-6105, Revision 2)

Validation (of a Product): The process of showing proof that the product accomplishes the intended purpose based on stakeholder expectations and the Concept of Operations. May be determined by a combination of test, analysis, demonstration, and inspection. (Source: NPR 7123.1)

Verification (of a Product): Proof of compliance with requirements/specifications. (Source: NPR 7123.1)