



Verification and Validation of SysML Models

Myron Hecht and Jaron Chen

***Presented to
System Engineering Forum
June, 2022***



Need for SysML Model Validation and Verification (V&V)

- Model Based Systems Engineering (MBSE) will not succeed without correct and complete models.
- Consequences of incomplete or incorrect models
 - *Integration failures due to erroneous or incomplete model interface blocks,*
 - *Violations of space weight or power constraints because of value properties leading in model block definitions and instantiations*
 - *Invalid analysis results because the model did not represent the system,*
 - *Inability to perform acceptance testing because requirements were not traced properly traced to the elements that satisfy them, and many others.*
- Net result: cost overruns and delays – just as in programs using conventional systems engineering practices.
- V&V methods should be integrated into programs using MBSE in order to avoid the same or worse program impacts



Previous Work

Model Transformation (transform SysML to another formalism such as Petri Nets that can be automatically analyzed)

- Manzoor Ahmad, Iulia Dragomir, Jean-Michel Bruel, Iulian Ober, Nicolas Belloir., "Early Analysis of Ambient Systems SysML Properties using OMEGA2-IFx," in SIMULTECH 2013, Reykjavik, Iceland, Jul 2013 .
- Yosr Jarraya and Mourad Debbabi, "Formal Specification and Probabilistic Verification of SysML Activity Diagrams," in IEEE Sixth International Symposium on Theoretical Aspects of Software Engineering, 2012.
- Messaoud Rahim, Ahmed Hammad, Malika Boukala-Iuoaleln, "Towards the Formal Verification of SysML Specifications: Translation of Activity Diagrams into Modular Petri Nets," in 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence, 2015.

Evaluation Methods (guided qualitative assessments of model content)

- Julie Fant, et. al., Systems Engineering Model Assurance Levels (MALs) Scale & Detailed Criteria Aerospace Technical Report, ATR-2020-00232-Rev A, March, 2021
- Edward R. Carroll and Robert J. Malins, "What Questions Would a Systems Engineer Ask to Assess Systems Engineering Models as Credible?", Sandia Report SAND20XX-XXXX, September 2020
- Dominique Ernadote, (Airbus Defense and Space Company), "A Framework for Descriptive Models Quality Assessment," IEEE 978-1-5386-4446-1/18, 2018.

Modeling and Style Guides (SysML language usage and diagram practices; not model content)

- Michael Vinnarcik and Heidi Jugovic, Digital Engineering Validation Tool Enables Efficiency Gains, <https://www.saic.com/blogs/digital-engineering-validation-tool-enables-efficiency-gains>: SAIC Corporation, 2021
- "System Modeling Standards and Guidelines," US Navy, Strategic Systems Program, Systems Engineering, System Modeling IPT, 28 December 2018
- Eric M. Lautenschlager and Michael Munoz, B-52 Model Based Systems Engineering (MBSE) Model Style Guide, MITRE Report MTR190557, September, 2019
- Georgia Tech Research Institute Digitally Integrated Systems Engineering Model Style Guide (draft), March, 2021
- Boeing Corp., CERP MBSE Style Guide, March 2021



Model V&V is Governed by Requirements

- Types of requirements
 - *Project Specific requirements*
 - Correctness of requirements
 - Completeness and accuracy of representation
 - Accurate traceability of requirements to design and to verification methods
 - Utility of produced artifacts (for development, management, design reviews, testing and verification, and sustainment)
 - Completeness and correctness of internal data, exports and imports
 - *Generic requirements*
 - Organization
 - Ease of navigation and information retrieval
 - Internal and External Documentation
 - Descriptive names
 - Complete diagrams
 - Correct use of SysML



Model Requirements Catalog

1. General:

- *Requirements for language (.e.g., SysML vs, UPDM)*
- *Incorporation of Government furnished profiles,*
- *Production of artifacts for model development, design reviews, testing and verification, and sustainment,*

2. **Requirements Diagrams and models:** amount and level of details needed for system requirements (very detailed requirements might be kept in a separate requirements management system)

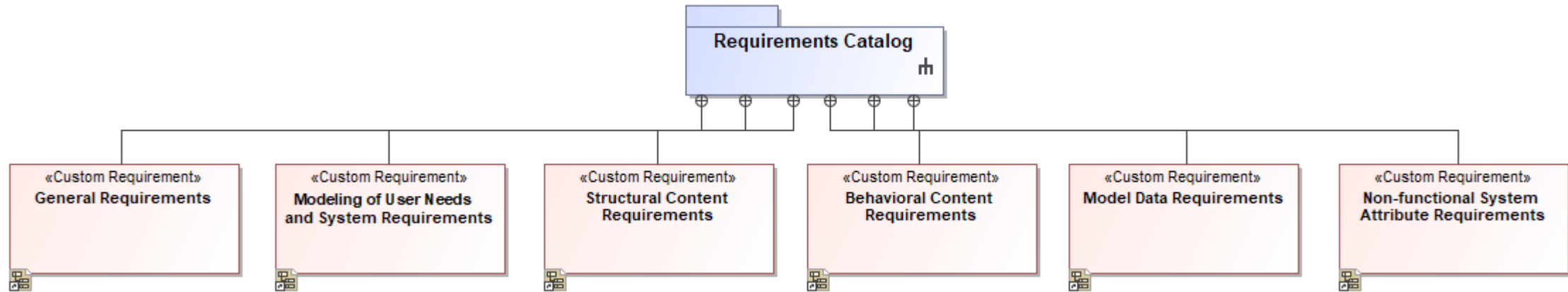
3. **Structural Model Elements and Diagrams:** Requirements for the amount and level of detail for structural content

4. **Behavioral Model Elements and Diagrams:** Requirements for the amount and level of detail for behavioral content.

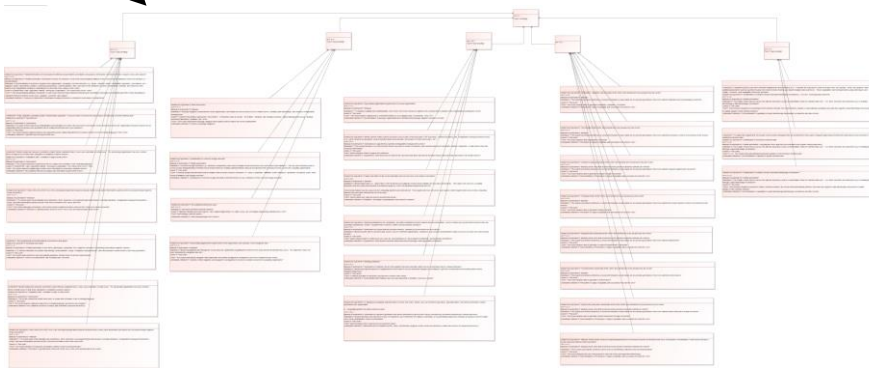
5. **Model Data**

6. **Modeling of non-functional system attributes:** Capacity and Response Time; Reliability, Maintainability, Availability; Safety; Cybersecurity

SysML Model Requirements Model (SyMREM)



Drill Down to detailed requirements



Implementation of Model Catalog within a SysML model



Attributes for SysML Modeling Requirements

| Attribute | Description |
|-------------------------|--|
| Name | Short name of the SysML model requirement |
| Requirements ID | Requirement label or number of the SysML model requirement |
| Requirements Text | The text of the of the SysML model requirement |
| Rationale* | Explanation and motivation of the SysML model requirement |
| Data to be Specified* | Additional data that must be supplied by the program |
| Verification Criterion* | The criterion (or criteria) which are used to determine whether the requirement has been satisfied |
| Manual or Automated* | Whether the requirement can be verified using an automated rule |
| Comment* | Additional information on the applicability and priority of the requirement |

*Customized fields for SyMREM

Requirements Table Fragment



| # | Name | Text | Id | Rationale | Detail to be Specified | Verification Criterion | Manual or Automated |
|----|------------------------------------|--|--------|---------------------------------|------------------------------------|-----------------------------------|----------------------------------|
| 1 | 1 General Requirements | Heading | 1 | | | | |
| 2 | 1.1 Model Development Requirements | Sub Heading | 1.1 | | | | |
| 3 | 1.1.3. Model Development tools | The model shall be captured using a fully supported based engineering (MBSE) tool which conforms to the modeling language of the model | 1.1.3. | In order for the model to be ma | Specific tool | Specified tool can open model f | Automated |
| 4 | 1.1.7. Conformance to Style Guide | The model shall conform to style and pattern guidelines defined in the model development plan | 1.1.7. | To enforce standards associate | Modeling style guide | Check for conformance with mc | Partial automation. Some items |
| 5 | 1.1.6. Descriptive Documentation | The documentation attribute or property of each model element shall contain text that provides descriptive and purpose information about the model (including all external referenced models) model views, diagrams, elements, and outputs | 1.1.6. | Documentation is needed to ex | What information is to be includ | Glossary or inspection of indivic | Partial automation. Automated |
| 6 | 1.1.5. Conformance with Model Dev | The model shall be developed in accordance with the model development plan. | 1.1.5. | To ensure basic model integrity | Items from DID or DD 1423 of t | Presence of specified items from | Manual |
| 7 | 1.1.2. Stereotypes and Profiles | The model shall apply stereotypes from model sponsor furnished profiles wherever the model uses related concepts | 1.1.2. | To ensure basic model integrity | Items from DID or DD 1423 of t | Presence of specified items from | Manual |
| 8 | 1.1.1. Modeling Language | The model shall be captured using in the modeling language specified by the program | 1.1.1. | The model should not violate ba | Validation suite – definition of h | No violations detected (overlap | Automated |
| 9 | 1.1.4. Absence of critical errors | The model shall not contain high severity errors detectable by automatic validation checks | 1.1.4. | The model should not violate ba | Validation suite – definition of h | No violations detected (overlap | Automated |
| 10 | 1.3 Data Imports and Exports | Sub Heading | 1.3 | | | | |
| 11 | 1.3.3 Automatic Updating | The model shall automatically incorporate updates to all referenced external data. | 1.3.3 | The SysML model may be used | Identification of referenced ex | Demonstration of capacity to u | Partial automation. Verifying th |



TABLE OF CONTENTS

Introduction 1

 Purpose 1

 Scope 1

 Overview 1

Requirements Catalog 2

 1 General Requirements 2

 1.1 Model Development Requirements 2

 1.1.1. Modeling Language 2

 1.1.2. Stereotypes and Profiles 2

 1.1.3. Model Development tools 3

 1.1.4. Absence of critical errors 3

 1.1.5. Conformance with Model Development Plan 3

 1.1.6. Descriptive Documentation 3

 1.1.7. Conformance to Style Guide 3

 1.2 Model Organization 3

 1.2.2. Package Diagrams 3

 1.2.3. Navigation Aids 4

 1.2.4. Package Names 4

 1.2.5. Naming across multiple related models 4

 1.2.1. Model Structure and Organization 4

 1.3 Data Imports and Exports 4

 1.3.1 Data Exports 4

 1.3.2 Data Imports 5

 1.3.3 Automatic Updating 5

 1.4 Views and Exports 5

 1.4.1. Model Development artifacts 5

 1.4.2. System Requirements Document 5

 1.4.3. Subsystem Design Document 5

 1.4.4. Software Design Document 5

 1.4.5 Verification, Validation, and Test Plans 6

 1.4.6 Test procedures 6

 1.4.7 Test reports 6

Requirements Catalog

1 General Requirements

Heading

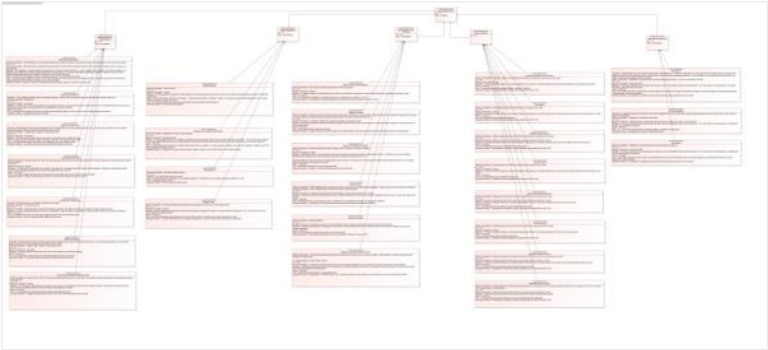


Figure 1. General

Hyperlinks:

 General

1.1 Model Development Requirements

Sub Heading

1.1.1. Modeling Language

The model shall be captured using in the modeling language specified by the program

Source:

This work

1.1.2. Stereotypes and Profiles

The model shall apply stereotypes from model sponsor furnished profiles wherever the model uses related concepts



Manual v. Automated Model Verification

- Manual V&V
 - *Evaluation of model's human meaning (semantics)*
 - Correctness of requirements allocation and verification
 - Completeness of model representation
 - Completeness and correctness of interfaces
 - Correctness of documentation
 - Correctness of value imports and exports
 - *Inspection and demonstration are the primary methods*
 - Test used for verification of quantitative results
- Automated V&V
 - *Evaluation of model's conformance to language rules and modeling conventions*
 - Requirements traceability
 - Structural and flow representations
 - Behavioral representations
 - *Scripts are the primary method of verification*
 - Analogous to static analyzers for software



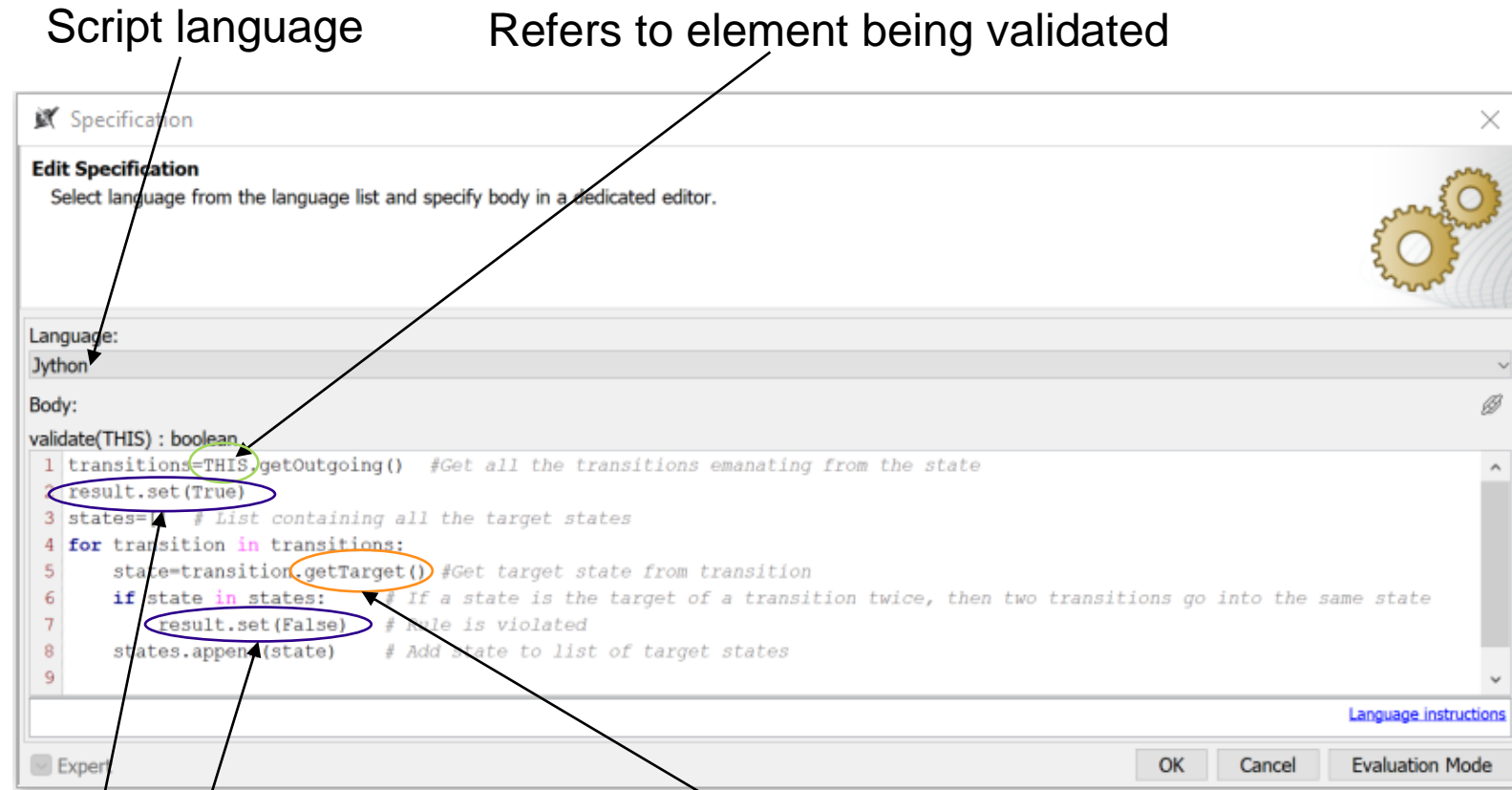
Examples of Manually Verified Requirements

- The model shall be organized in a consistent manner (e.g. by organization, by hierarchy, or by subsystem)
 - Verified by inspection
 - Rationale: large models are difficult or impossible to understand unless their organization is clearly understood
 - Additional details to be specified in requirement: how model should be organized
- The model shall include package diagrams that capture and describes the model organization
 - Verified by inspection
 - Rationale: package diagrams are the primary means of depicting model organization
 - Additional details to be specified: how package diagrams should be organized
- The model shall include diagrams that depict links and enable navigation to all diagrams and views contained in the model
 - Verified by demonstration
 - Rationale: model navigation aids enable information to be found that might otherwise be missed
 - Additional details to be specified: what navigation aids should be provided



Example of Automatically Verifiable Requirement

The model shall have a different target state for each state/transition pair on the state machine diagram




























Setting Boolean output to determine if validation rule is violated or not

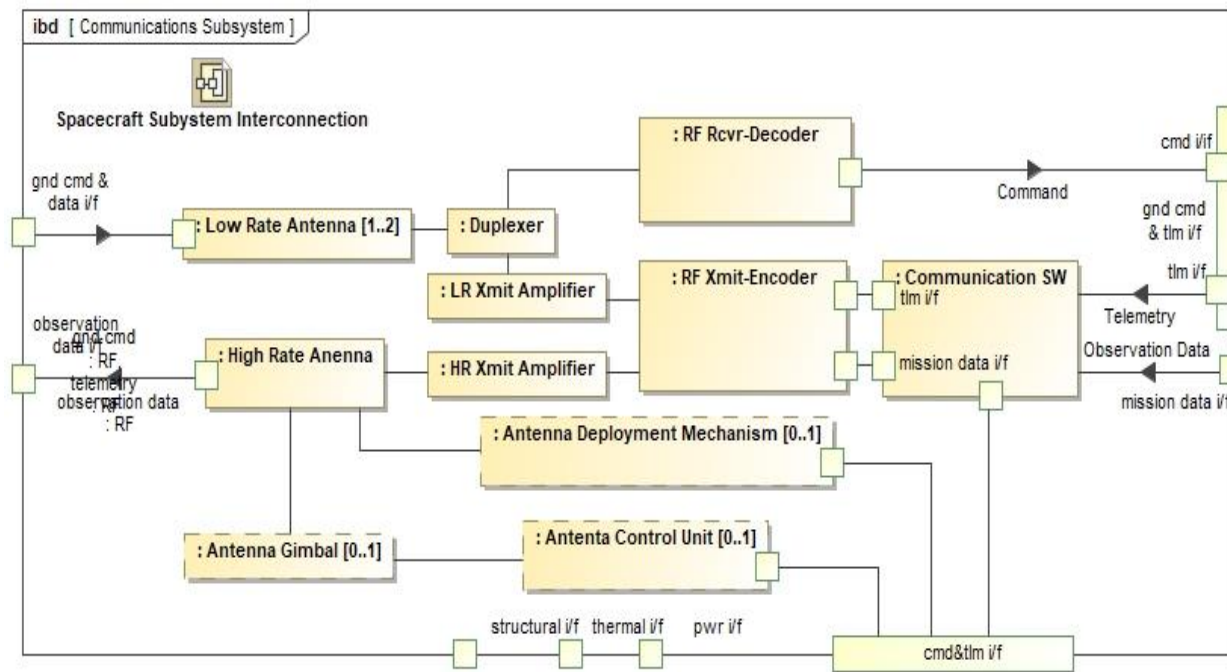
Method from MagicDraw API that allows access to element properties

Automated Validation Suite

146 Validation Rules written for Cameo Systems Modeler

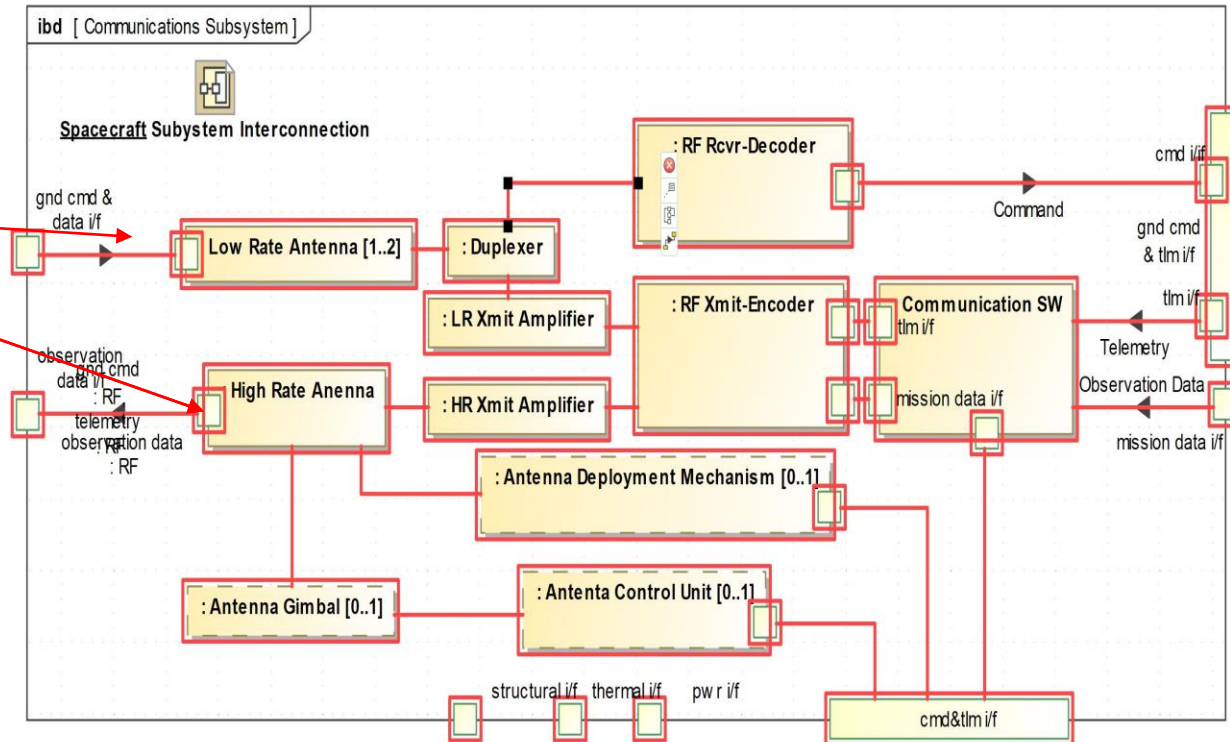
Element type: ... Scope (optional): {jwy} ... Filter:

| # | Name | ▽ Constrained Element | errorMessage |
|----|----------------------------------|---|--|
| 9 | { } TRANSITIONSOURCE |  Transition | No operation owns an output parameter typed by the signal (or its general classifier) that triggers |
| 10 | { } TRANSITIONTRIGGER |  Transition | All transitions (except those exiting connection points or pseudostates) must have triggers. |
| 11 | { } TRANSITIONTRIGGERFLOW |  Transition | This transition is triggered by a signal but is not associated with any item flows or flow sets. |
| 12 | { } TRIGGERFLOWMISMATCH |  Transition | The signal triggering this transition is not conveyed on any related item flows or flow sets. |
| 13 | { } CONTEXTPARTS |  System context [Class] | System context blocks must own at least one part property. |
| 14 | { } STATE_MACHINE_ERROR_HANDLING |  StateMachine | The model shall have state charts that address error conditions and alternative paths. |
| 15 | { } STATEOWNER |  StateMachine | State machines must be owned by blocks. |
| 16 | { } STMINTEGRITY |  StateMachine | State machines may only call operations owned within their owning block's structural decomposition |
| 17 | { } EVENT_STATES |  State | The model shall have a new state for each state/event pair on the state chart. |
| 18 | { } STATE_ENTRIES_AND_EXITS |  State | The model shall have states with at least one entry and at least one exit. |
| 19 | { } STATE_NAMING |  State | States must have unique names. |
| 20 | { } STATEDOCUMENTATION |  State | All states must have documentation. |
| 21 | { } STATENAME |  State | States must be named. |
| 22 | { } STATEREACHABILITY |  State | All states must have at least one incoming transition. |
| 23 | { } SUBMACHINECONNECTIONS |  State | States that are submachines must have all entry and exit points associated with connection points. |
| 24 | { } SOFTWAREFUNCTION | «» software [Class] | This software element does not own any operations. |
| 25 | { } SIGNALEVENTSIGNAL |  SignalEvent | Signal Events must have a signal defined. |
| 26 | { } SIGNALDOCUMENTATION |  Signal | All signals must have documentation. |
| 27 | { } SIGNALNAME |  Signal | All signals must be named. |
| 28 | { } SENDINCOMING |  SendSignalAction | If incoming object flows to a Send Signal event are realized by an item flow or flow set, the signal c |
| 29 | { } SENDSIGNALMATCH |  SendSignalAction | The signal sent by a send signal action must match the signal typing its input pin. |
| 30 | { } SENDSIGNALPORTMATCH |  SendSignalAction | The assigned and inferred ports using an item flow realization must match. |
| 31 | { } REQT_HIERARCHY |  Requirement [Class] | Requirements should be in a hierarchy. |
| 32 | { } REQT_LINKS |  Requirement [Class] | Requirements must be in a satisfied relationship with another element or needs a derived relationsh |
| 33 | { } REQT_RATIONALE |  Requirement [Class] | The rationale for all requirements shall be provided in the model using the document property of th |
| 34 | { } REQT_SATISFY_VERIFY |  Requirement [Class] | All requirements have either (1) associations to non-requirements or (2) SysML contain, derive refir |



Check that all connections are through ports

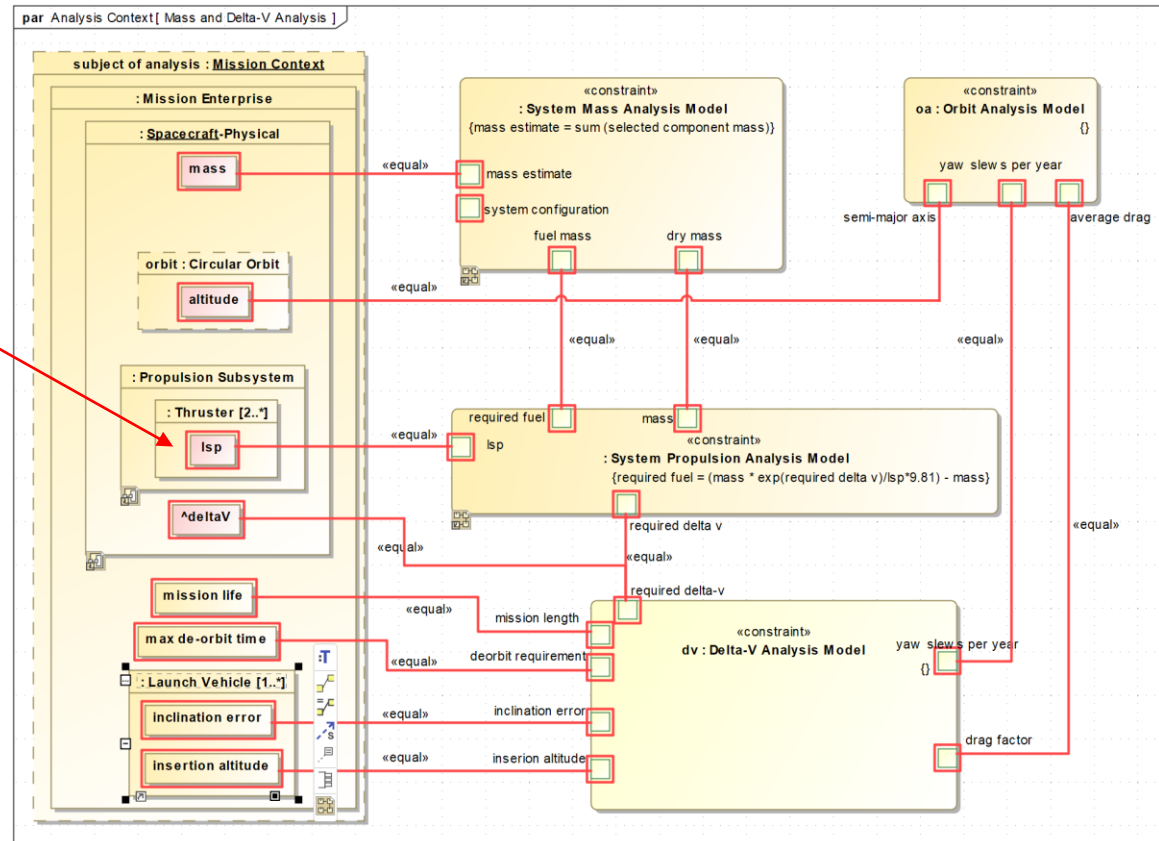
2 of many violations:
Element: Connector



Typing of Value properties

Error: “The model shall include data type for all attributes and properties”

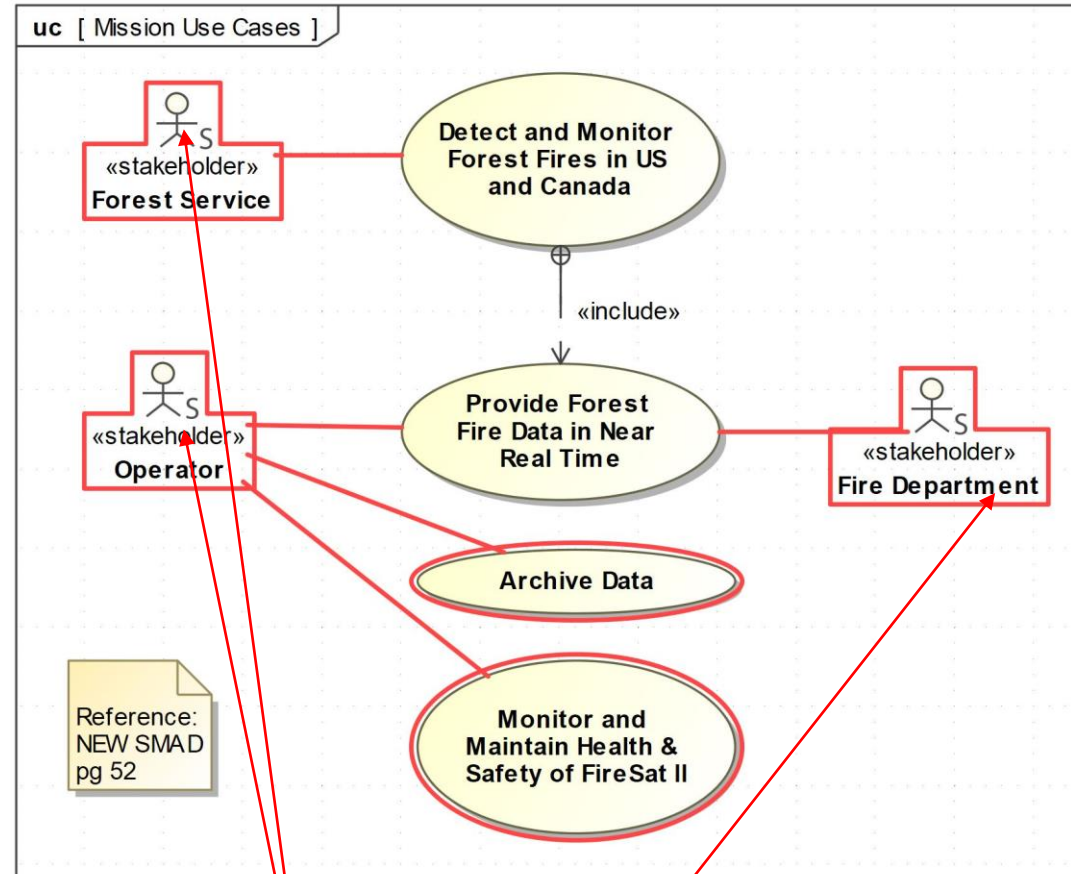
Element: required delta-v





Documentation of Model Elements

Element: Actor
(Operator)



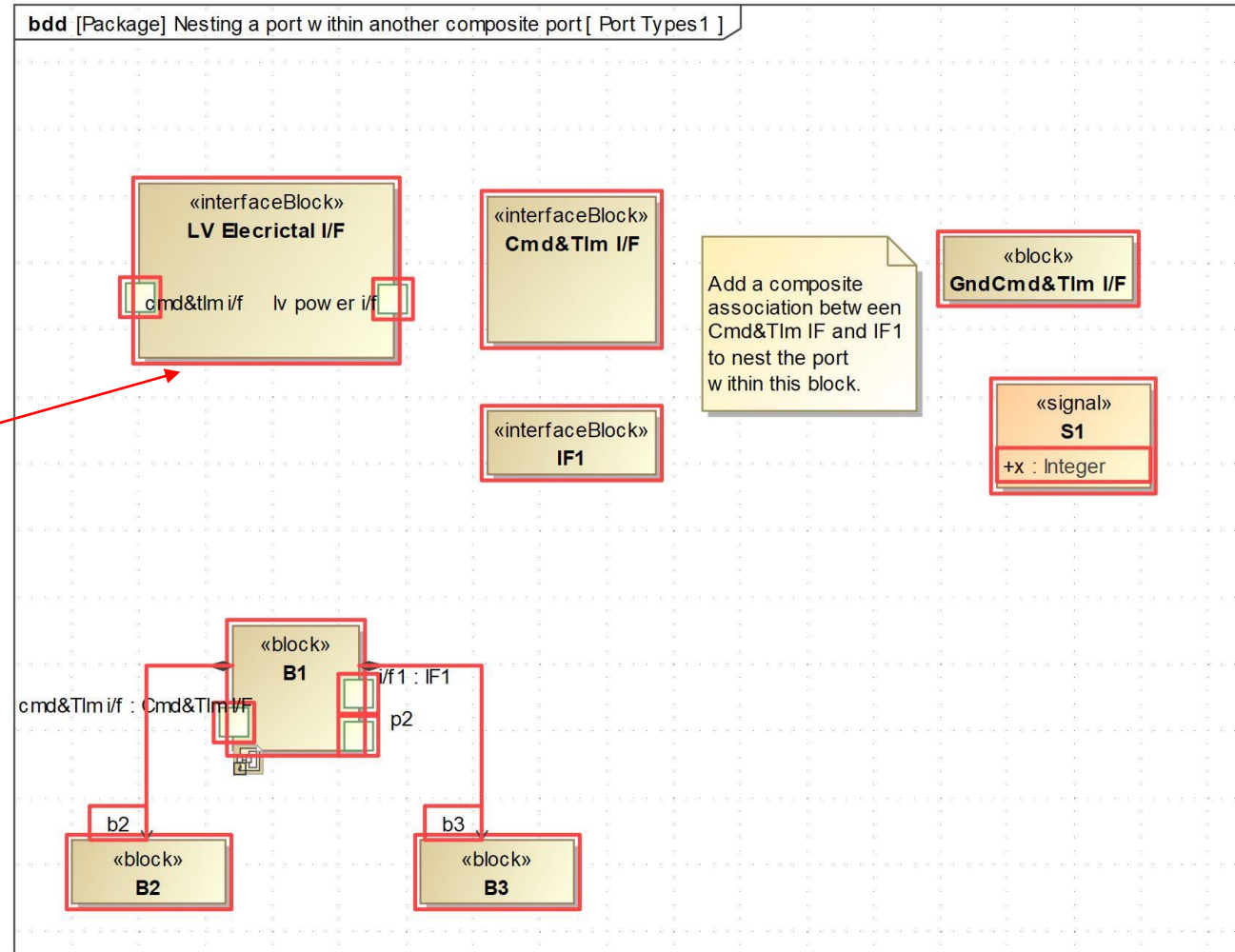
Error: "All actors need documentation."



Traceability between blocks and requirements

Error: “All blocks representing system components shall be traceable to a requirement.”

Element: SysML Blocks





Conclusions

- A necessary condition for MBSE and for Digital Engineering is that the models are correct
- Model Requirements are necessary for model verification and validation (V&V)
 - *Model requirements are distinct from system requirements.*
- V&V includes manual and automated methods
 - *Manual methods are primarily inspection and validation, and are necessary to validate model requirements on aspects of the model that are interpreted by humans (natural language content, model navigation, and model output)*
 - *Automated methods can be used to check model syntax, relationships, and structure*
- A catalog of candidate requirements and verification methods in both document and model form will soon be available to assist in development of model requirements



Note about automated validation examples

- The diagrams on charts 14-17 are from the model accompanying the book entitled *Architecting Spacecraft with SysML* (2017) written by S. Friedenthal and C. Oster. The authors contributed the model to the public domain under a BSD-2 at license. It is available at <http://www.sysml-models.com/spacecraft> and was used for this work with their permission
- The model was intended to illustrate how a simplified MBSE methodology using SysML can be applied to architect a system, but it was not intended to be comprehensive, complete, or rigorous. As a result, there are many gaps in terms of missing elements, documentation, and other details which the validation rules detected. For example, only selected requirements were flowed down from mission to component level to illustrate requirements traceability in the model, and there are many other mission and system requirements that are not explicitly satisfied or verified. In addition, many of the ports are left untyped, and the documentation fields are not provided for each model element.
- However, the model is useful for demonstrating how the validation rules can be used to detect and ultimately improve the completeness and rigor of the model, and ensure consistency with an organization's modeling practices.
- We gratefully acknowledge both the effort and expertise that were invested in its creation and the generosity of the authors in allowing its use for this purpose.



Myron Hecht:

Myron Hecht is a Senior Project Leader at The Aerospace Corporation where he specializes in MBSE and in reliability, safety, and systems engineering for satellites, ground control systems, and other complex weapons systems. He has been supporting programs in MBSE and SysML since 2017, and also teaches the subject at the Aerospace University, UCLA, and the Air Force Nuclear Weapons Center. He also is a consultant to the Nuclear Regulatory Commission in reactor safety and control systems and a lecturer at the UCLA School of Engineering and Applied Sciences. He has authored more than 100 refereed publications in model-based systems engineering, reliability, safety, and products liability. Myron holds a B.S. in Chemistry, an M.S. in Nuclear Engineering, an M.B.A., and a J.D. degree all from UCLA.

Jaron Chen:

Jaron Chen is a Senior Member of the Technical Staff at the Aerospace Corporation. He works in in the areas of Model Based System Engineering (MBSE), machine learning, software tools development, discrete event simulation, and integration of modeling techniques in support of space and ground communications systems. He holds an M.S. in Computer Science from Carnegie Mellon University and a B.S. from the University of California Irvine.