International Council on Systems Engineering
A better world through a systems approach

SysML v1 to SysML v2 Model Conversion Approach

July 29, 2025

Frank Salvatore, SAIC

Sanford Friedenthal, SAF Consulting

Work sponsored by Director of Digital Engineering, Modeling and Simulation within the DoD Office under Secretary of Defense for Research and Engineering (OUSD(R&E) and the Naval Air Systems Command

INCOSE International Symposium 2025
| Ottawa, Canada

SysML v2 Status

- SysML v2 beta specifications (i.e., KerML, SysML v2, Systems Modeling API & Services) approved by the OMG on June 30, 2023 and are now in the finalization phase
- Final adopted specifications for KerML, SysML v2, Systems Modeling API & Services approved by the OMG on July 21, 2025. Announcement can be found here. https://www.omg.org/sysml/sysmlv2/
- You can check out the press release here Object Management Group Approves Final Adoption of the SysML V2 Specification.

Purpose

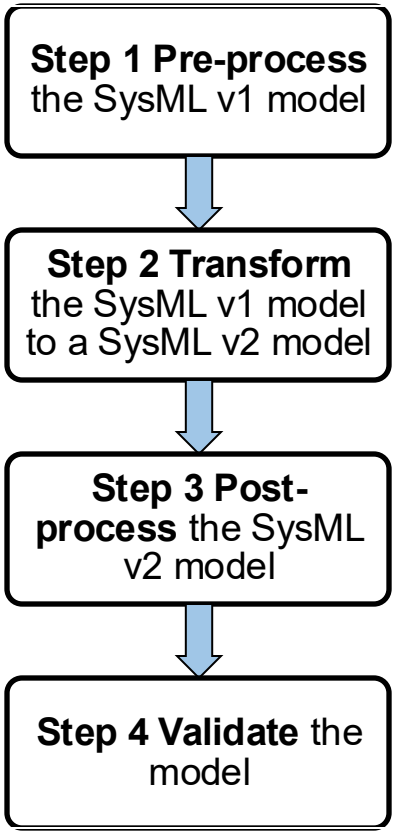
- Summarize results from converting a SysML v1 model to a SysML v2 model
  - Documented in a Model Conversion Guidance report
- Part of the SysML v1 to SysML v2 Transition Guidance Project sponsored by the Director of Digital Engineering, Modeling and Simulation within the DoD Office under Secretary of Defense for Research and Engineering (OUSD(R&E)

Transition Project Background

- The SysML v1 to SysML v2 Transition Guide Project
  - Initiated in December 2022
  - Goal: Provide guidance on how an organization and project can transition their MBSE efforts from SysML v1 to SysML v2
  - Deliverables:
    - https://www.omgwiki.org/MBSE/doku.php?id=mbse:sysml_v2_transition

Model Conversion Process

1. Pre-process the SysML v1 model
   - Ensure model conforms to SysML v1 specification
   - May need to remove customizations, ensure model is well-formed, and adheres to standard guidelines (e.g., naming conventions)
2. Transform the SysML v1 model to a SysML v2 model
   - Tool automation executes standard transformation specification
3. Post-process the SysML v2 model
   - Reorganize, refactor, and refine model to leverage SysML v2 capabilities
4. Validate the model
   - Evaluate how well it satisfies model objectives and update as needed



Model Conversion Example

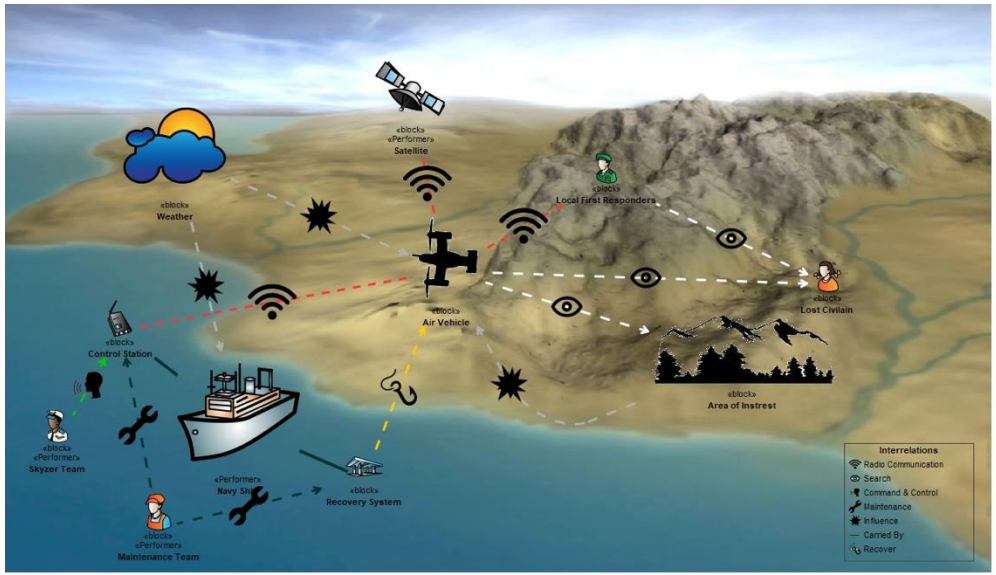
- Skyzer Mission Model
  - Developed by the Systems Engineering Research Center (SERC) (citation included on *reference slide #25*)
  - SysML v1 model in Cameo
  - Mission: UAV launched from a ship to perform a search and rescue mission
  - Approximately 5300 elements
  - Includes 6 of 9 diagrams and req'ts table
    - does not include par, req, stm
  - Includes packages, dependencies, blocks, attributes, parts, ports, connections, associations, use cases, actors, activities, actions, swim lanes, control flows, object flows, lifelines, messages, requirements, constraints, and trace and satisfy relationships
    - does not include proxy and full ports, interface blocks, states, transitions, test cases, and derive and verify relationships, constraint blocks, constraint properties, and binding connections, …

Skyzer v1 Model

pkg [Skyzer Mission Model]

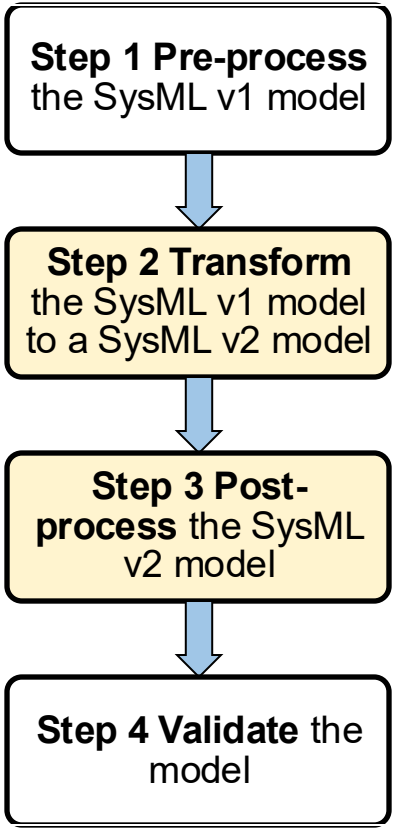
Skyzer Mission Model

- 0. Mission Statement
- 1. Mission Requirements
- 2. Mission Structure
- 3. Mission Use Cases
- 4. Mission Behavior
- 5. Mission Parametric
- 6. Mission Interface Definitions
- 7. Skyzer UAV
- 9. Support Elements
- 10. Lessons Learn
- Skyzer Mission Model
- Start



Model Conversion Approach [1 of 3]

- Used pilot implementation (Jupyter Lab with Plant UML) to manually create the SysML v2 model
- Performed steps 2 and 3 using an incremental approach
  - Transform the SysML v1 model to a SysML v2 model
  - Post-process the SysML v2 model
- Transform sufficient portion of the model to demonstrate approach and gain lessons learned



Model Conversion Approach [2 of 3]

- The SysML v1 model was manually transformed to a SysML v2 model. This was done based on modeler's understanding of the specification and the transformation rules.

SysML v1 Constructs

- Package structure
- Blocks and parts
- Ports and connectors
- Value properties and value types
- Requirements and requirement hierarchy
- Use cases
- Activities
- Interactions (sequence diagrams)
- Requirements relationships
- Stereotypes

Model Conversion Approach [3 of 3]

- The model was post processed to include significant reorganization and refactoring
    - Reorganize package structure
    - Part definitions in part definitions package
    - Refactor parts hierarchy
    - Refactor parts interconnection
    - Action definitions in action definitions package
    - Refactor action hierarchy
    - Integrate behavior (use cases, actions, sequence/messages)
    - Refactor the requirements hierarchy
    - Refactor requirements traceability

Model Conversion Package Structure

SysML v1

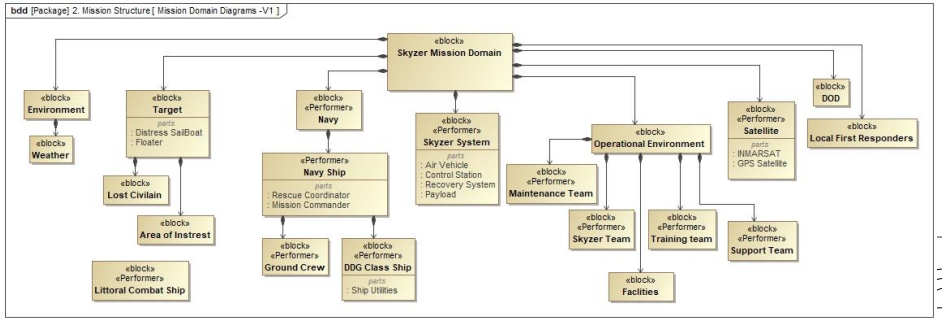
```
package SkyzerMissionModel_transformed{
    package '0.MissionStatement'{↔}
    package '1.MissionRequirements'{↔}
    package '2.MissionStructure'{↔}
    package '3.MissionUseCases'{↔}
    package '4.MissionBehavior'{↔}
    package '5.MissionParametric'{↔}
    package '6.MissionInterfaceDefinitions'{↔}
    package '7.SkyzerUAV'{↔}
    package '9.SupportElements'{↔}
    package '10.LessonsLearned'{↔}
    package '11.LanguageCustomization'{↔}
}
```

SysML v2 Transformed

```
package SkyzerMissionModel_refactored{
    package Definitions{
        package PartDefinitions{↔}
        package ItemDefinitions{↔}
        package AttributeDefinitions{↔}
        package RequirementDefinitions{↔}
        package UseCaseDefinitions{↔}
        package ActionDefinitions{↔}
    }
    package <'9'> SupportElements {↔}
    package <'11'> LanguageCustomization{↔}
    package Mission_Domain_Level{↔}
    package System_Level{↔}
    package RequirementsAllocations{↔}
}
```

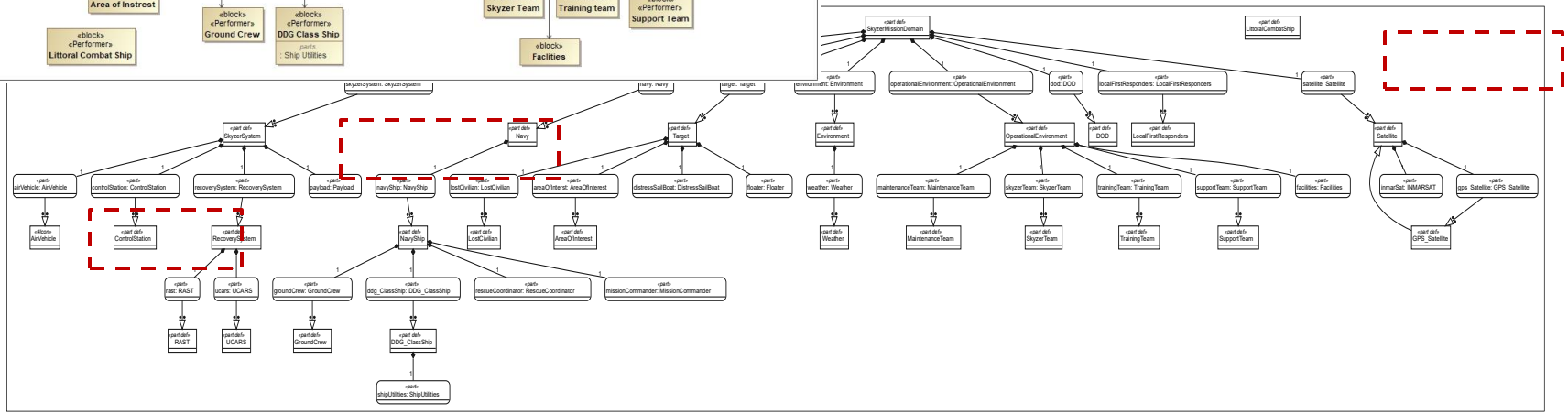
SysML v2 Refactored

Parts Hierarchy – SysML v1 to SysML v2 Transformed



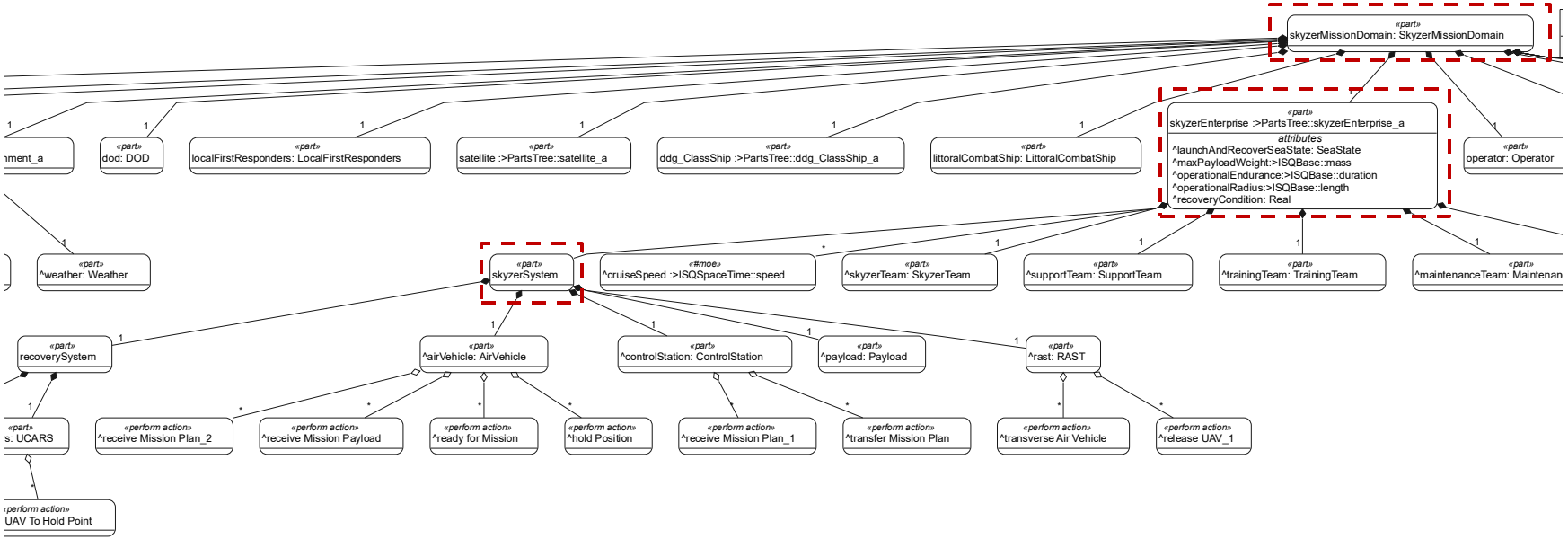
SysML v1

SysML v2 Transformed



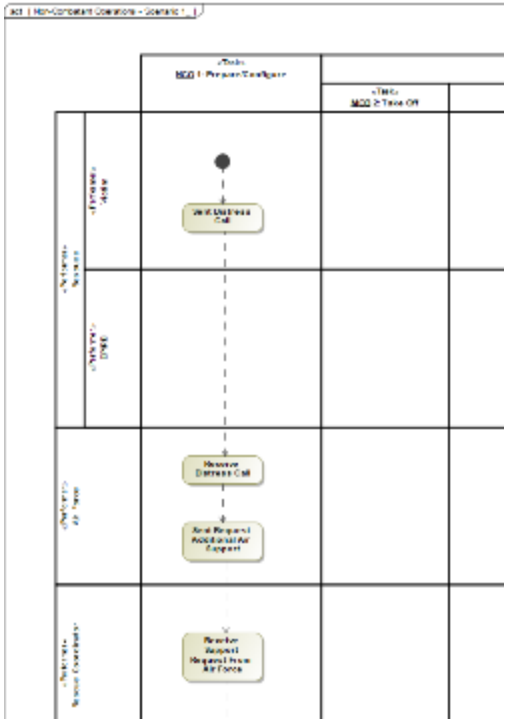
Mission Part Def to SkyzerSystem Part to SkyzerSystem Part Def

Parts Tree – SysML v2 Refactored

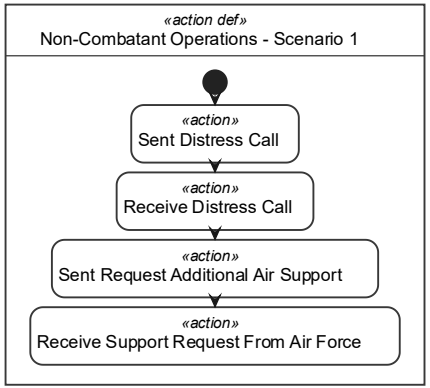


Mission Part to Enterprise Part to SkyzerSystem Part

Behavior – SysML v1 to SysML v2 Transformed



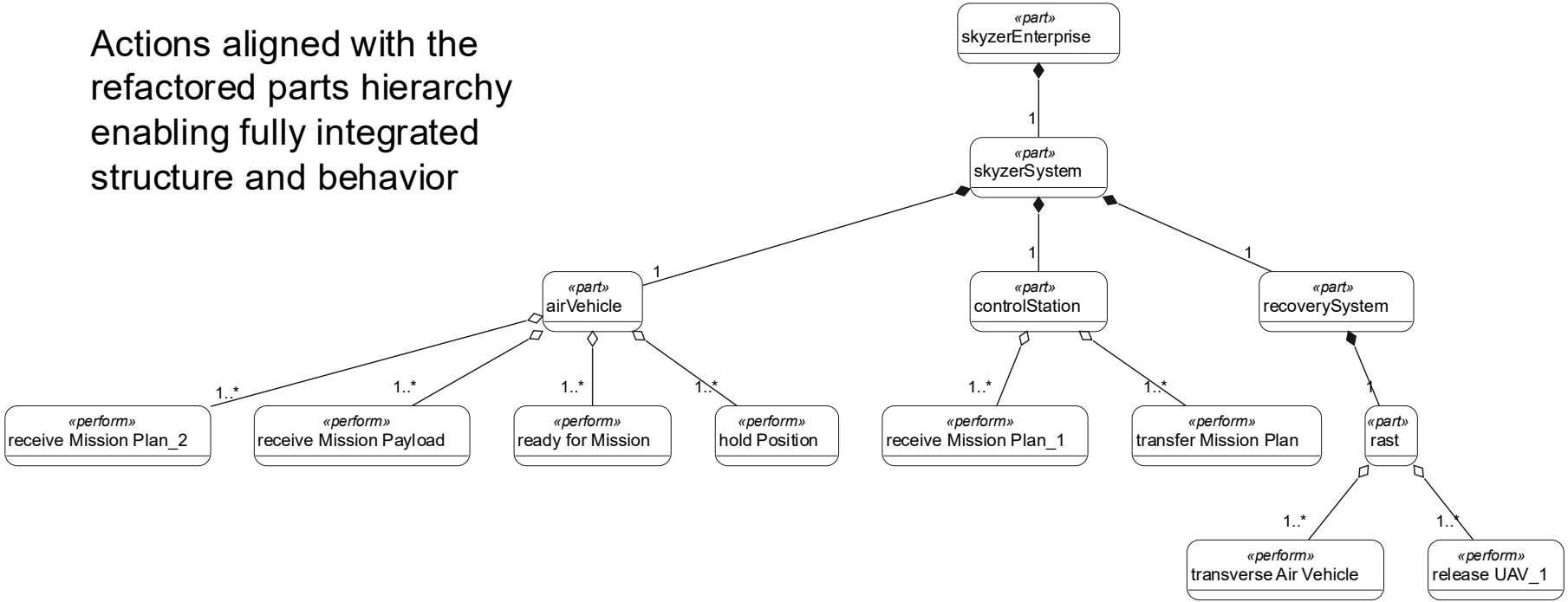
SysML v1



SysML v2- Transformed
Removed Swim Lanes
(relate actions to parts in a later step)

Behavior – SysML v2 Refactored Actions Performed by Skyzer System

Actions aligned with the refactored parts hierarchy enabling fully integrated structure and behavior



Requirements – SysML v1 to SysML v2 Transformed

10	1.1.2	R 1.1.2 Imaging Capability	The system shall perform various imaging capabilities for search and rescue
11	1.1.2.1	R 1.1.2.1 Image Processing	Ground Control to have image-processing capabilities to support identifying locations, lost personnel, and drop locations for search and rescue
12	1.1.2.2	R 1.1.2.2 Image-Exploitation	Ground Control to have image-exploitation capabilities to support imagery mapping and seeking capabilities

SysML v1

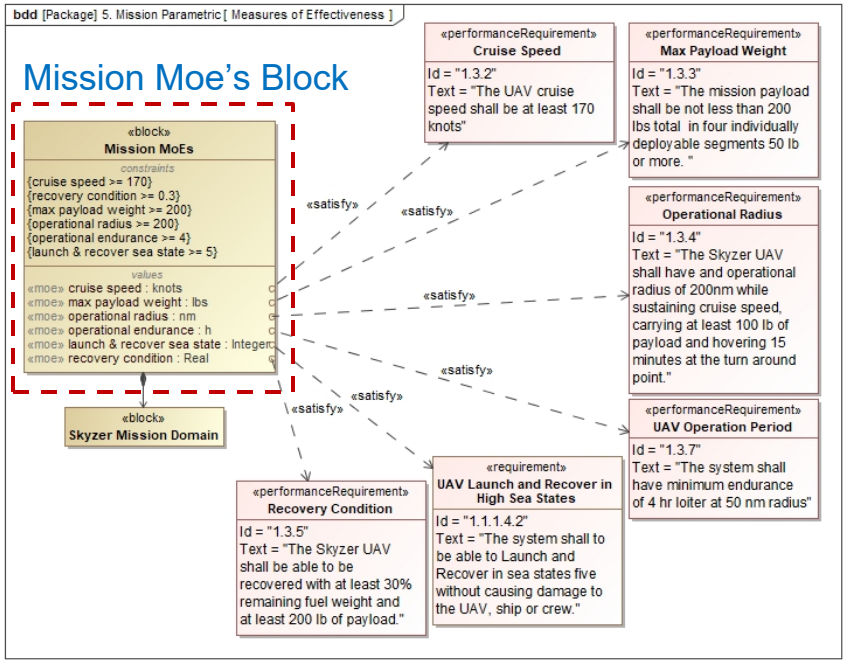
```
requirement def ImagingCapability{
    doc /* The system shall perform various imaging capabilities for search and rescue*/
    requirement def ImageProcessing{
        doc /* Ground Control to have image-processing capabilities to support identifying
        locations, lost personnel, and drop locations for search and rescue */
    }
    requirement def ImageExploitation{
        doc /* Ground Control to have image-exploitation capabilities to support imagery
        mapping and seeking capabilities */
    }
}
```

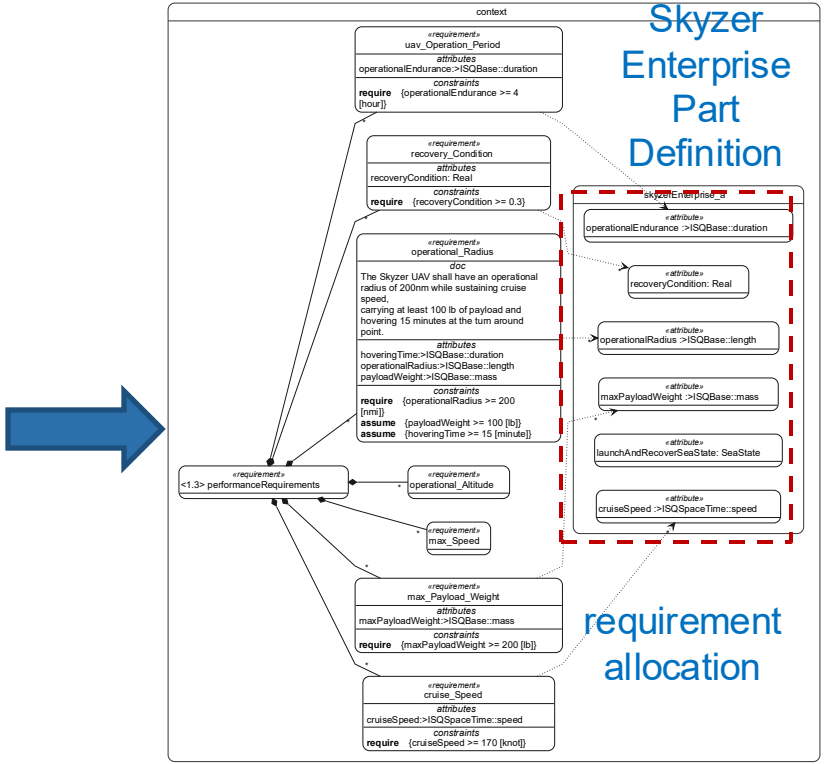
SysML v2 Transformed

Requirements – SysML v2 Refactored Specification with Integrated Requirements Hierarchy

```
package MissionSpecification{
    requirement <'1'> missionSpecification{
        requirement <'1.0'> missionRequirements{
            requirement airworthiness;
            requirement imagingCapability:ImagingCapability{
                requirement imageProcessing:ImageProcessing;
            }
            requirement uas_ControlSegment;
            requirement uav_Capabilities{↔}
        }
        requirement <'1.1'> operationalRequirements {↔}
        requirement <'1.2'> functionalRequirements{↔}
        requirement <'1.3'> performanceRequirements{↔}
        requirement <'1.4'> designConstraints{↔}
    }
```

Requirements Satisfaction/Allocation SysML v1 to SysML v2 Refactored





SysML v1

SysML v2 Refactored

Requirements – SysML v2 Refactored Example of Performance Requirement with Constraint

```
requirement operational_Radius{
    doc /*The Skyzer UAV shall have an operational radius of 200nm while sustaining cruise speed,
    carrying at least 100 lb of payload and hovering 15 minutes at the turn around point.*/
    attribute operationalRadius :> ISQ::length;
    attribute payloadWeight :> ISQ::mass;
    attribute hoveringTime :> ISQ::time;
    require constraint {operationalRadius >= 200 [nmi]}
    assume constraint {payloadWeight >= 100 [lb]}
    assume constraint {hoveringTime >= 15 [minute]}
}
```

Some Refactoring Observations

- Integrated SysML v2 part hierarchy integrates SysML v1 block decomposition with swim lanes in activities, lifelines in sequence diagrams, actors in use cases, and blocks in OV-1
- Integrated SysML v2 behavior integrates use cases, actions flow, and message sequence
- Need to refactor the requirements allocations/satisfactions based on the revised structure and behavior
- Selected requirements may be formalized with constraints to make them more precise
- Many opportunities to leverage SysML v2 to further refine the model (e.g., short name and aliases, stakeholder concerns, metadata, …)

Summary

- Converting a SysML v1 model to a SysML v2 model should include pre-processing, transformation, post-processing and validation
- Conversion process should be performed incrementally and validated at each step
- Post-processing, which includes reorganizing and refactoring the model, is needed to leverage SysML v2 capabilities and benefits
- The benefit is a much more integrated and precise model

References

- Skyzer Model
  - Developed by the Systems Engineering Research Center (SERC) under Task Order 0802 with the Naval Air Systems Command, under contract HQ0034-19-D-0003 with Office of the Under Secretary of Defense for Research & Engineering
    https://ime.sercuarc.org/alfresco/mmsapp/mms.html#/login
    - User: openmbeeguest
    - PW: guest
- Comparing SysML v2 with SysML v1 by S Friedenthal
  Presentation to SE DSIG on March 21, 2023 (omg doc # syseng/2023-03-02)
- SysML v2: Highlighting the Differences with SysML v1 by S Friedenthal and E Seidewitz PPI SyEn Edition 123 April 2023
  https://www.ppi-int.com/systems-engineering-newsjournal/ppi-syen-123/
- Report: SysML v1 to SysML v2 Model Conversion Approach
  https://www.de-bok.org/asset/5bd90f82fab101bdf093103f76ce74ec5411300e

Thank You!!

35th Annual INCOSE international symposium

hybrid event

Ottawa, Canada
26 - 31 July 2025