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

A Framework for Seamless Interoperability

Tara Sarathi
tara.sarathi@ll.mit.edu

Ricardo Martinez
rmartine@mathworks.com

INCOSE International Symposium 2025 | Ottawa, Canada

Drivers Behind the DoD’s Shift Toward Digital Engineering

Evolving Mission Complexity

- Joint operations increasingly involve coordination across multiple domains (space, land, air, etc)
- Mission scenarios shift rapidly, requiring the ability to reconfigure and re-evaluate system capabilities on short timelines

System-of-Systems Integration Challenges

- Modern platforms (e.g., sensors, comms, etc) must function as part of a dynamic system-of-systems
- Siloed tools and inconsistent data standards prevent clear traceability between mission needs and technical implementation

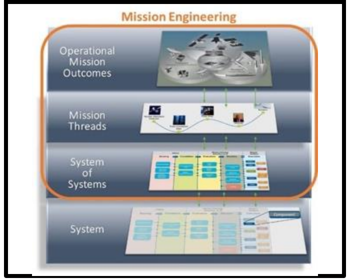
Need for Data-Driven Decision Making

- Operational environments are unpredictable; decisions must be grounded in objective data
- Accelerated acquisition timelines require early insight into how design choices impact mission effectiveness

Digital Engineering enables the speed, insight and integration that today’s missions demand

The Vision: Integrated Digital Engineering Environment

Mission Engineering



Reusable MBSE Model Libraries

Collaborative Cross Domain Modeling And Simulation

Digital Twins

AI/ML Capabilities

Integrated Digital Engineering Environment

A connected digital environment linking mission requirements, system models, and simulation enables continuous verification and comprehensive assessment of mission effectiveness

AI: Artificial Intelligence
ML: Machine Learning

Challenges of Linking Mission to System to Simulation Models

Toolchain Fragmentation

- Mission and System models often reside in Cameo, while high-fidelity simulations are often built in MATLAB/Simulink, creating disconnected workflows
- While the two tools can natively be integrated, passing more complicated parameters between the two is challenging

Limited Interoperability and Data Exchange

- Current SysML v1 standards lack dynamic data linking needed for seamless traceability
- Simulation outputs have limited integration back into system or mission models, requiring manual data handling and custom coding

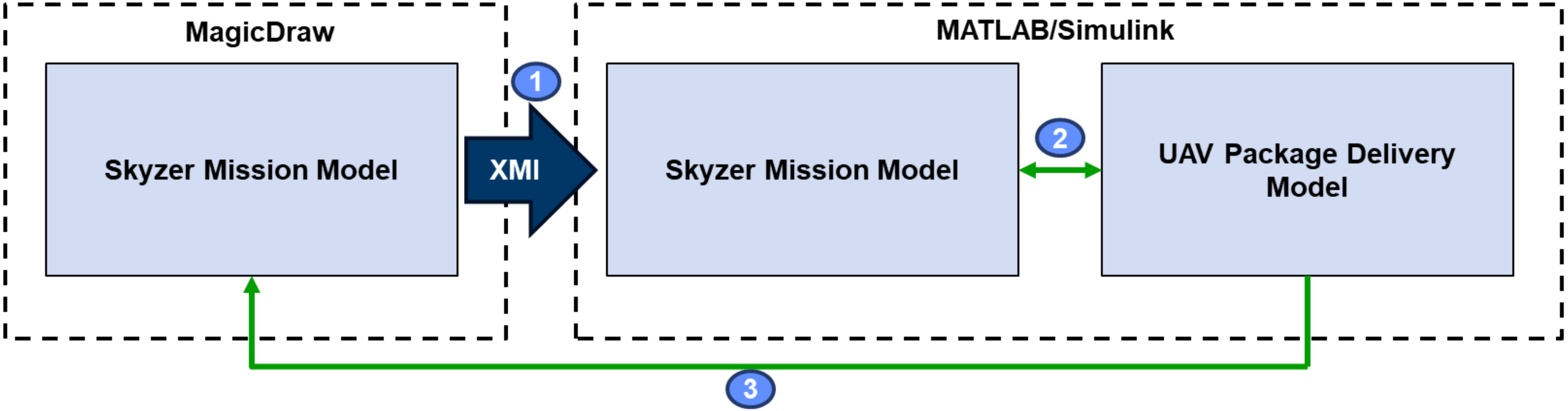
Managing Model Variations and Traceability

- Multiple system design variants exist, but linking each variant's simulation results back to mission requirements is difficult
- Lack of bidirectional traceability impedes automated verification and slows decision making processes

We developed a framework that automates mission requirement flow through system models to simulations, enabling continuous verification and comprehensive mission assessment

Integrated Framework Overview

User Story: As the mission planner, I need to understand which variant of the Skyzer system will best satisfy mission constraints (e.g. speed, weight, safety). By linking the mission model, system variants, and simulation data, I can perform rapid, automated assessments of which configuration is most effective for a given mission



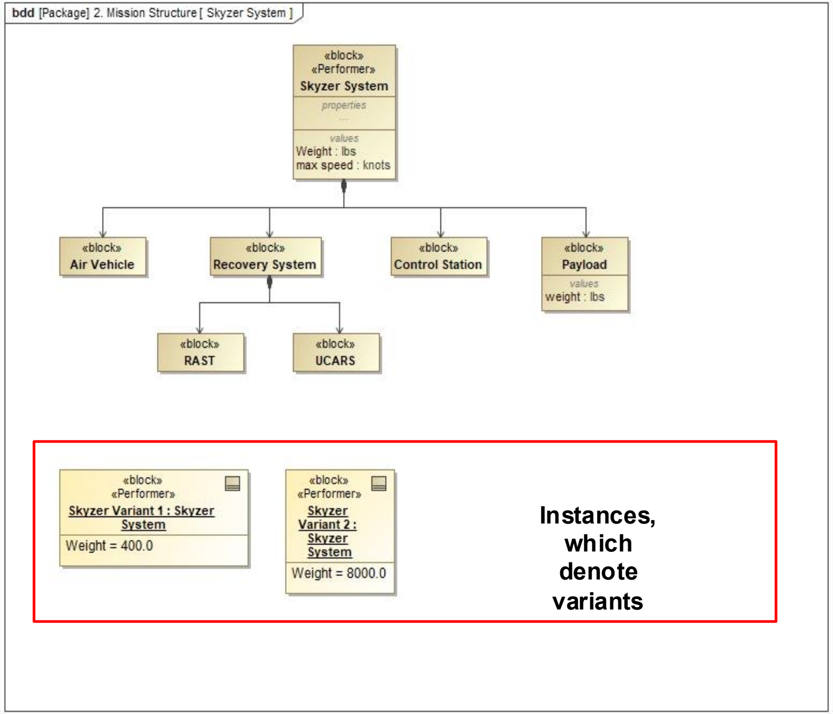
1. Import to System Composer:
   - System Requirements
   - System Diagrams
   - Interface Definitions
   - System Parameters
   - System Variants

2. Link to High-Fidelity Models:
   - Relevant environmental parameters
   - Relevant system Parameters
   - Relevant variants

3. Report to MBSE model in MagicDraw:
   - Recorded simulation results

Model Overview: Skyzer Mission Model

- **Serves as the foundation for linking mission-level objectives to system-level architecture and behavior**
- **Leverage open-source model used by OMG for SysMLv1 to SysMLv2 transition efforts**
- **Original open-source model was modified to include:**
  - Variant options for key system parameters
  - Additional mission requirements to align with simulation model
  - Test cases for verification of mission requirements



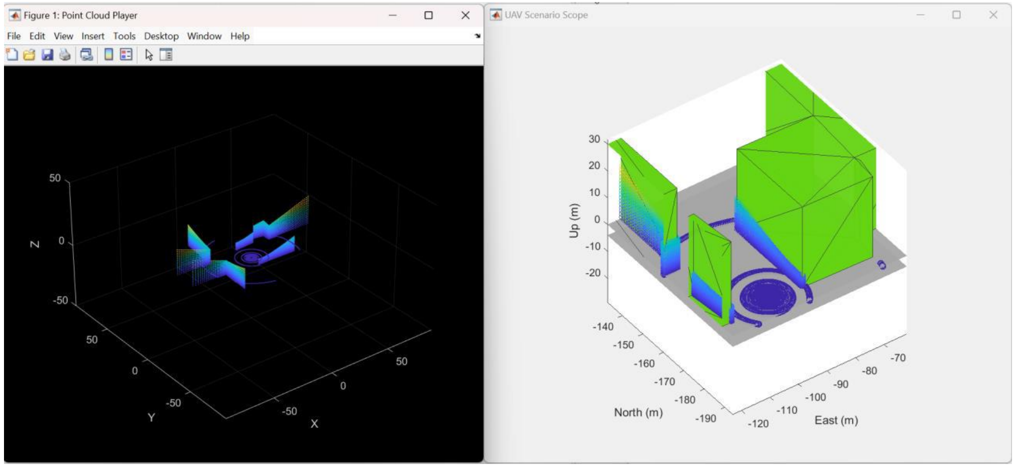
Model Overview: Skyzer Mission Model

#	△ Name	Text	Satisfied By	Verified By
1	1.2.1.7 UAV High Sea States Operati	The Skyzer system shall be able to execute the in-flight portion of its mission in Sea State 7 conditions with wind aloft not less than 80 knots	Wind Speed : Skyzer Mission Mo	Sea State During Missior
2	1.2.2.1 Fly Autonomously	The Skyzer UAV shall be able to fly Autonomously		Successful Takeoff and L
3	1.2.2.2 Autonomously navigate to se	The Skyzer UAV shall Autonomously navigate to search area specified in mission plan		Successful Takeoff and L
4	1.2.2.8 UAV Fly Patterns	The system shall autonomously fly a search pattern specified in the mission plan		Successful Takeoff and L
5	1.3.1 Max Speed	The UAV shall have a max speed of 200 knots	max speed : Skyzer Mission Mod	Speed During Mission
6	1.4.3 Weight	The Skyzer UAV shall weigh no more than 5,000 lb.	Weight : Skyzer Mission Model::9	Weight During Mission

Skyzer Mission Requirements added for demonstration

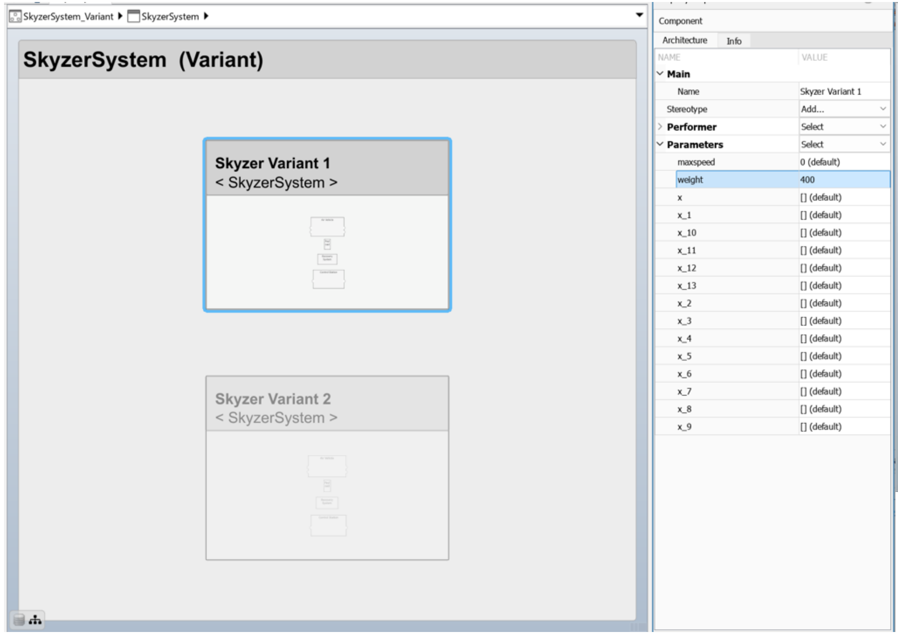
Model Overview: UAV Package Delivery Model

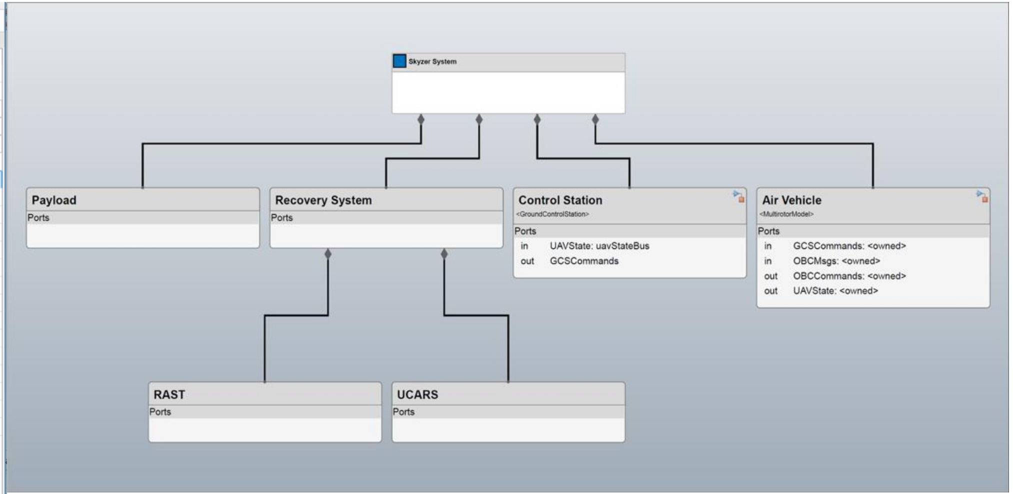
- The UAV Package Delivery model provides a representation of high-fidelity models likely to be used for analysis and verification.
- Originally developed by MathWorks, it serves as an example for users interested in implementing a small multicopter simulation.
- The model includes realistic flight dynamics, control systems, and test cases for performance evaluation.
- Selected due to its mission profile aligning closely with that of the Skyzer Mission Model, minimizing the need for extensive modifications.



Linking Models to Simulation

- Initial step is to bring in system model into System Composer via XMI
- This creates a representation for establishing digital thread

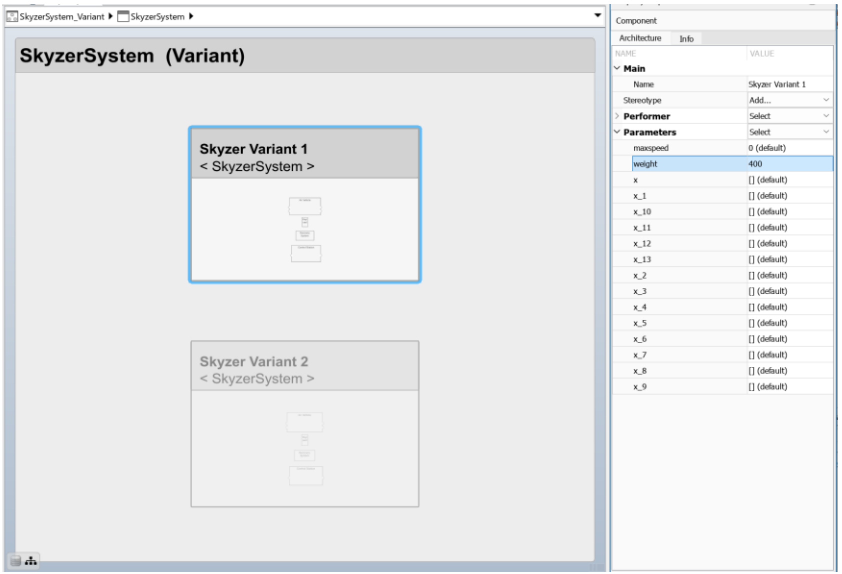


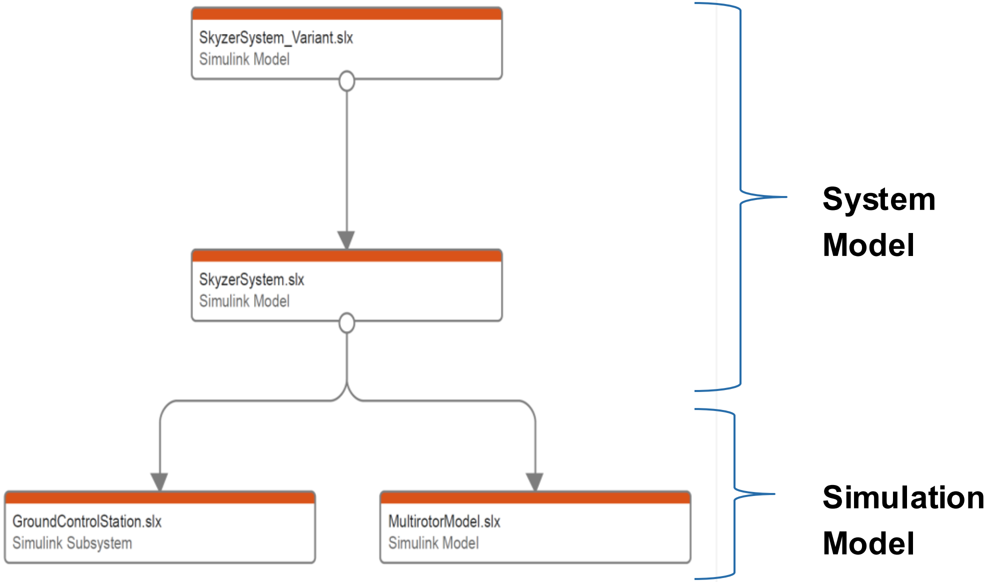


System Composer representation of Skyzer System model with variants

Linking Models to Simulation

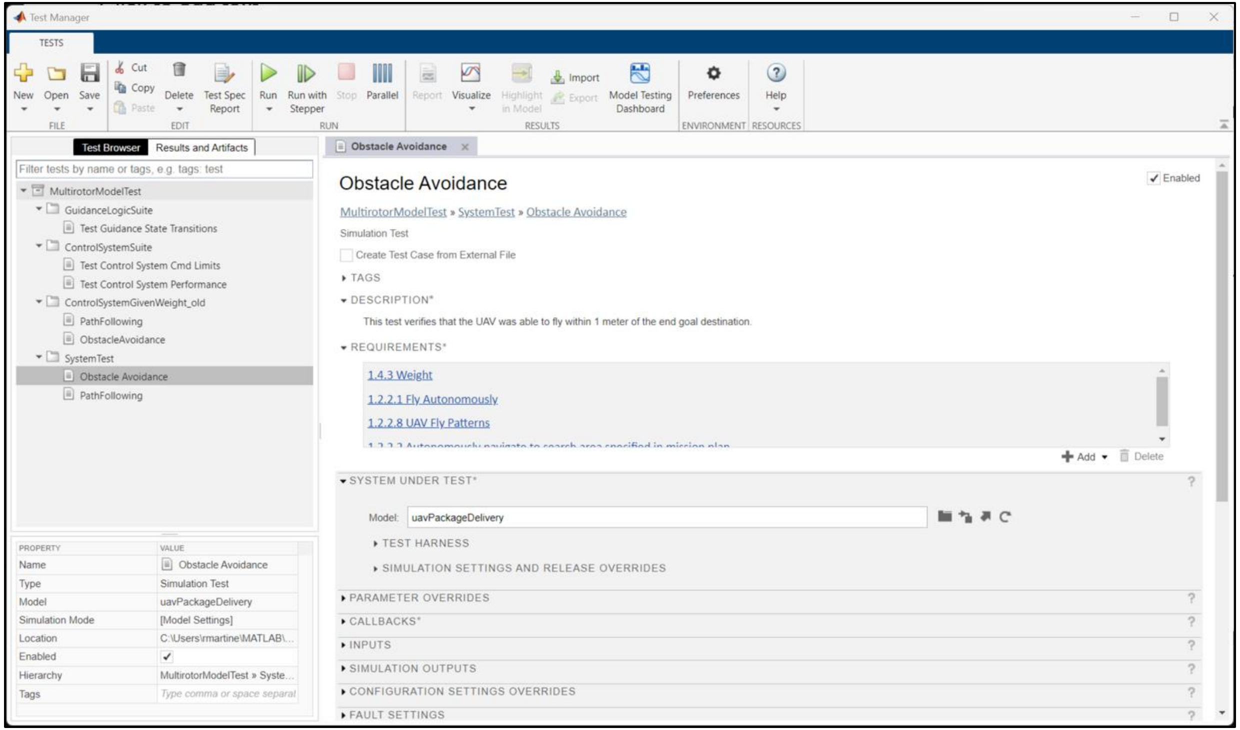
- Next is to link system model (Skyzer) to simulation model (UAV Package Delivery)
- This is done by link the system model elements directly to their respective Simulink components





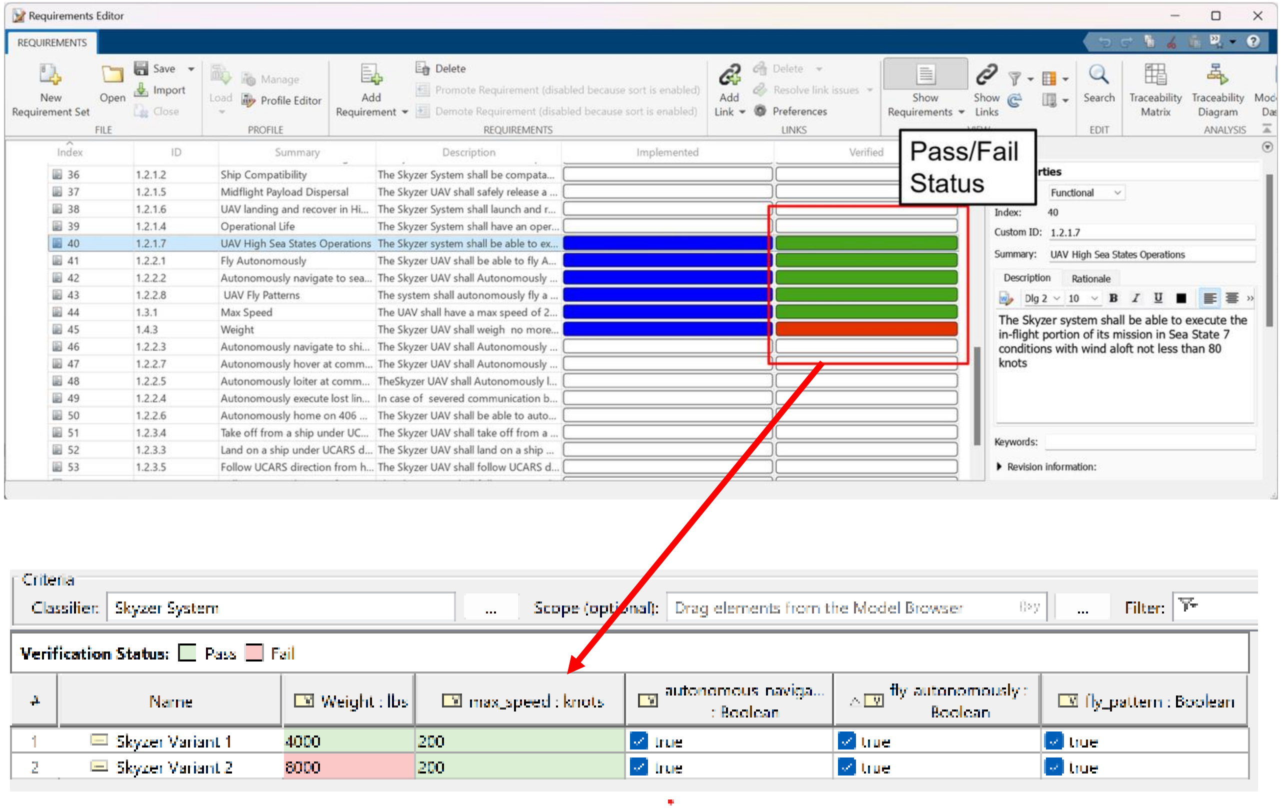
Verification and Capturing Results

- The UAV Package Delivery Model includes test cases that assess performance against predefined criteria.
- These test cases were mapped to system requirements from Skyzer Mission model
- Tests can be executed for each system variant configuration



Test Case Definition Example

Verification and Capturing Results



- Results from tests cases now passed back into MagicDraw for traceability
- A custom utility uses XMI identifiers to update the Skyzer model with pass/fail status, maintaining synchronization with.

Conclusion

- Developed a framework that links mission models, system architectures, and simulations to enable seamless end-to-end traceability
- Demonstrated how this integration supports continuous verification of mission requirements across system variants
- Enables faster, data-driven decision making by connecting operational intent to technical performance