Satellite Co-Orbital Engagement Engineering with System Composer

Authors: Justin Yu, Miron Lev
Advisors: Oleg Yakimenko, Mike Anthony
Contributors: Manny Martinez, Ossi Saarela, Becky Petteys

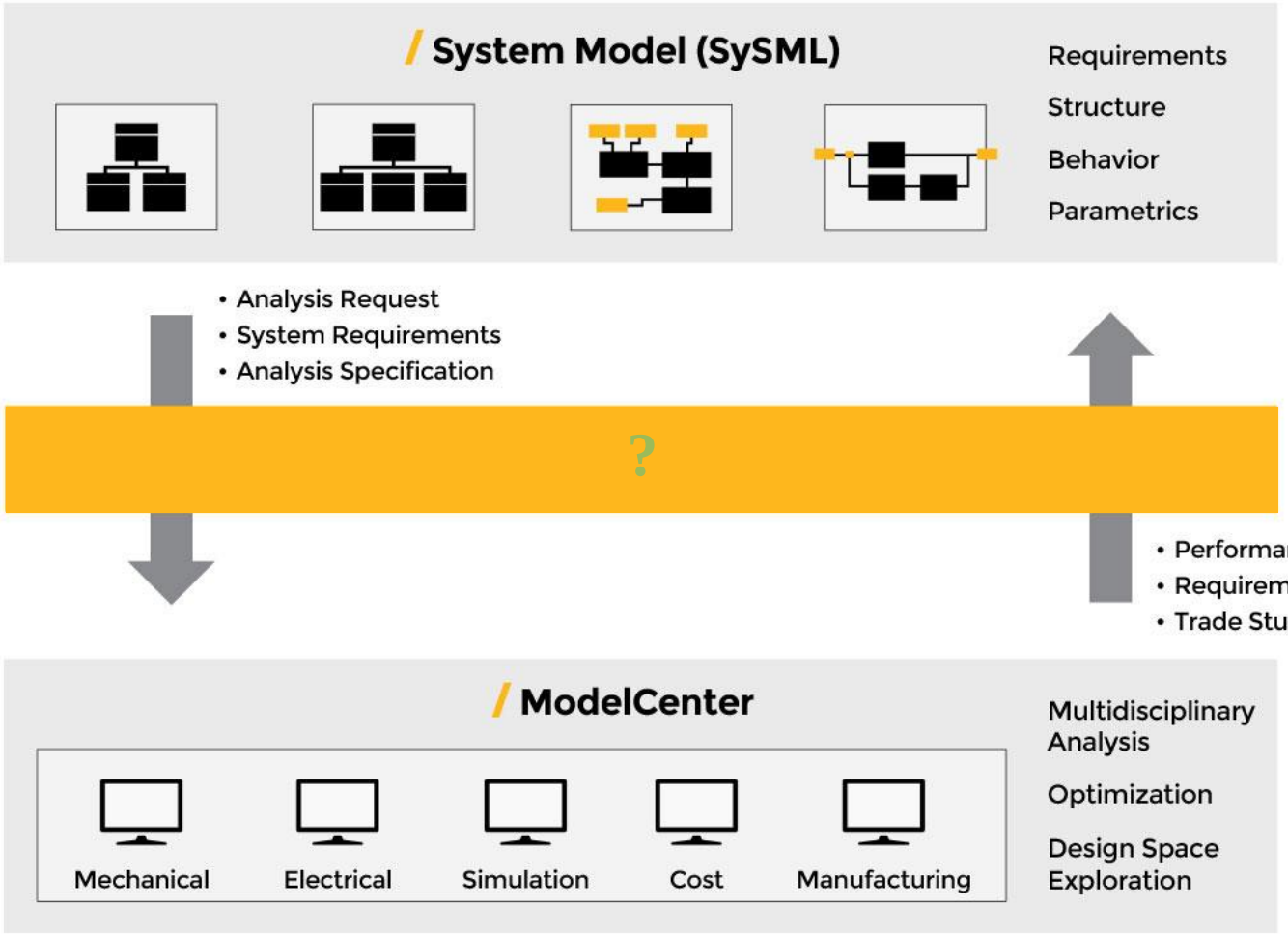
OUTLINE

- Background
- Representative Mission to be Modeled
- Mission Modeling Domain
- Integration with Cameo System Modeler
- Integration with MathWorks' System Composer
- Demonstration of Model Execution
- Conclusions

Section Header

BACKGROUND

CONNECTING SYSTEMS ENGINEERING DOMAIN WITH DOMAIN ENGINEERING MODELS



Authoritative Source of Truth (an integrated set of engineering models)

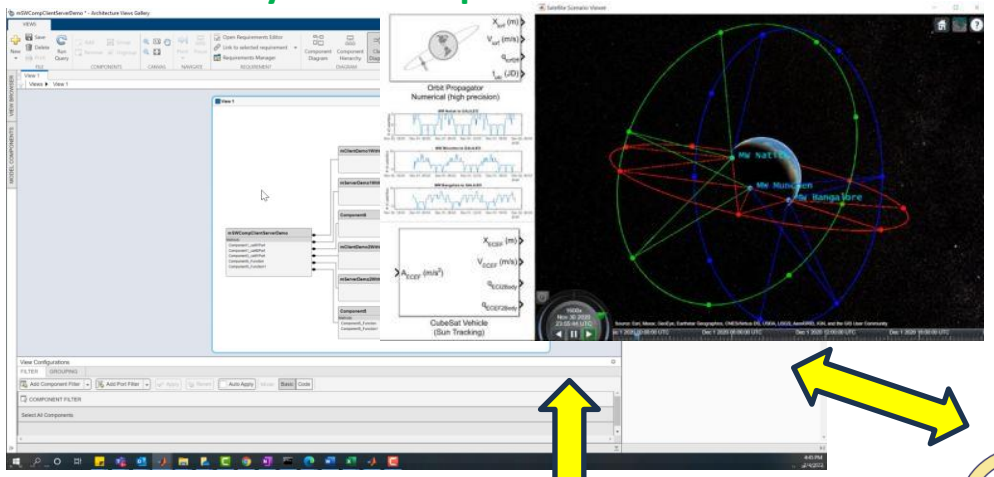
The centerpiece of the digital transformation is the ASoT

The System Model is the glue that ties together disparate models into a single ASoT

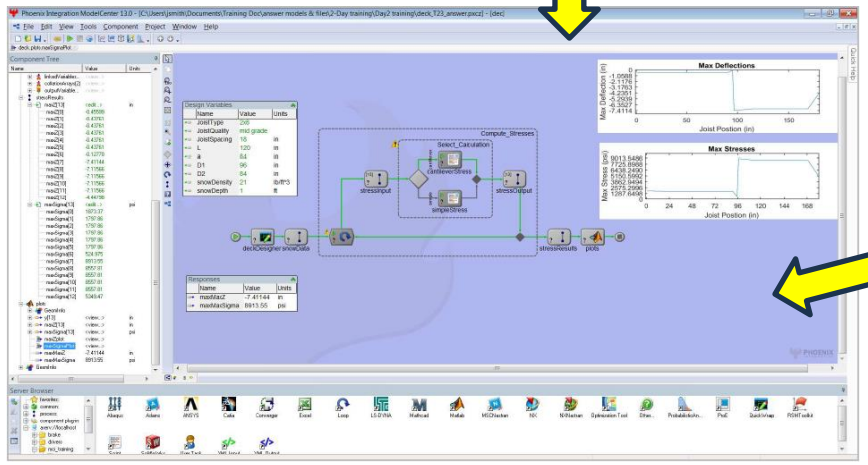
The ASoT resides in a digital ecosystem (interconnected infrastructure, environment and methodology)

DIGITAL ECOSYSTEMS ERA

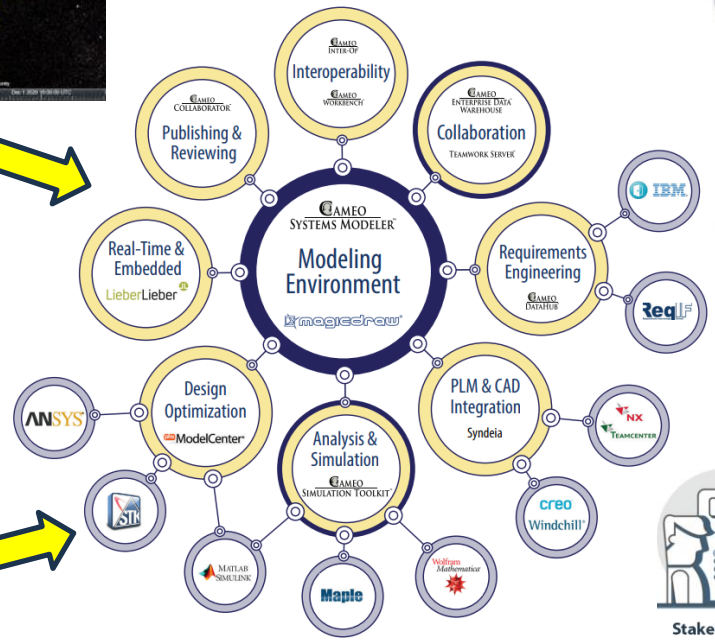
MathWorks System Composer



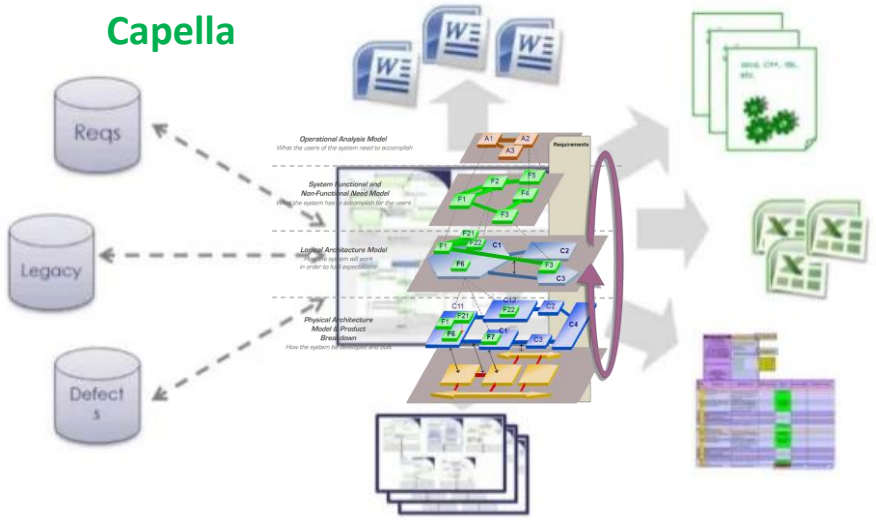
Ansys ModelCenter



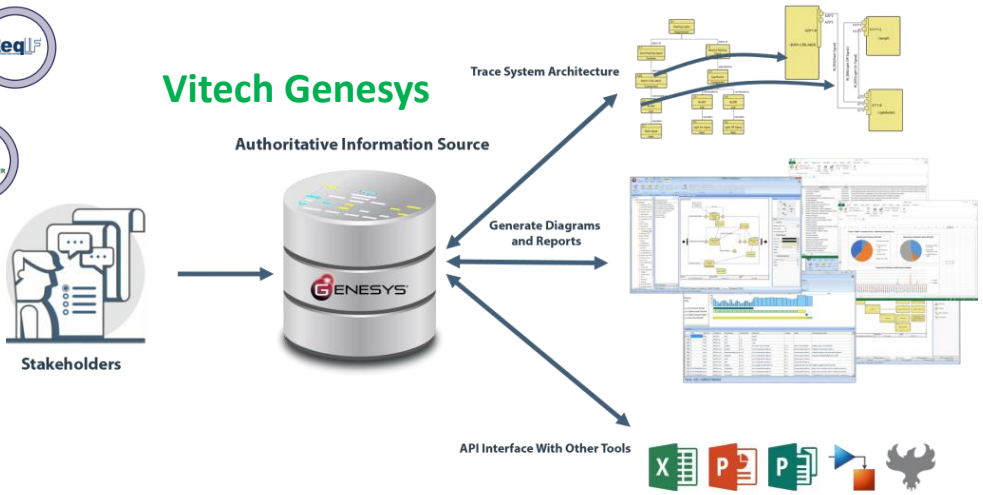
Cameo Systems Modeler



Capella

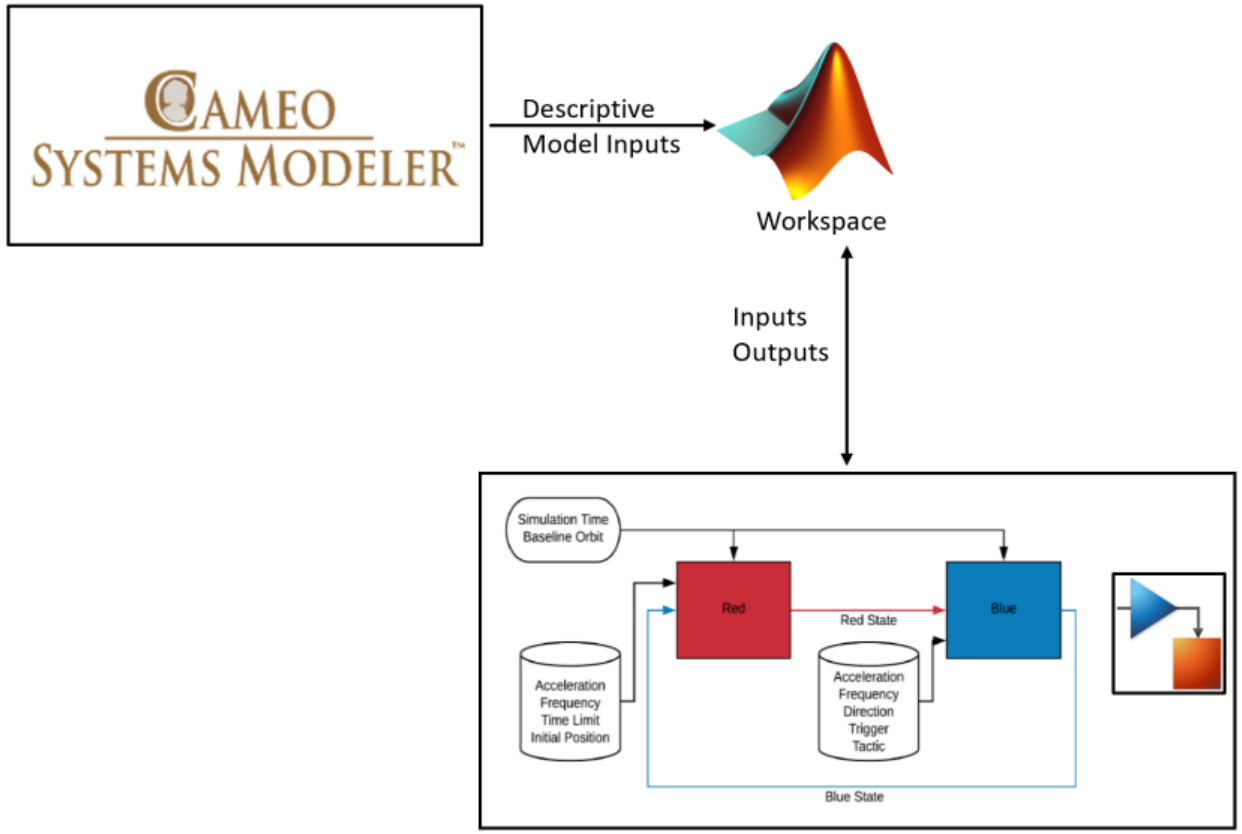


Vitech Genesys



PREVIOUS INTEGRATION ATTEMPTS

- Simulink high-fidelity simulation was integrated with the descriptive model of the spacecraft engagement mission developed in Cameo System Modeler (CSM)
- The value of MBSE approach was clearly demonstrated via
  - enabling enhanced mission analysis
  - facilitating tactics assessments
  - validating operational requirements
  - providing outputs to derive new requirements
- Full integration was not achieved cased by application programming interface incompatibility
  - Incompatible value properties between CSM/Simulink
  - GUI limitations
  - Manual requirements in execution
- Another attempt included integrating Cameo System Modeler mission model with rewriting spacecraft engagement mission model with high-fidelity satellite engagement model converted to Python



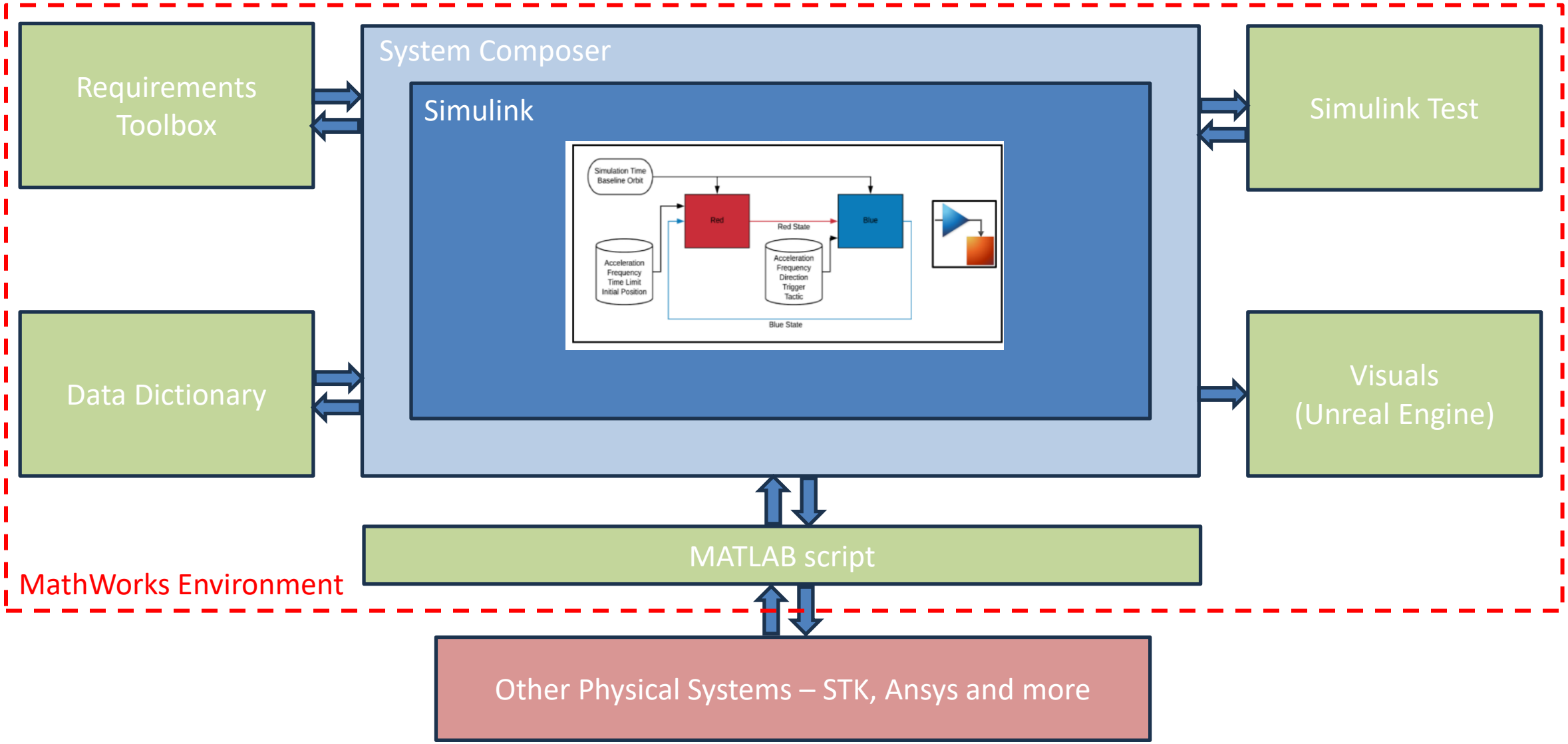
Rangel, D., Pavalkis, S., and Yakimenko, O. “Incorporation of a High-Fidelity Simulink Model into the Cameo Systems Modeler Environment for a Conceptual Design of Satellite Engagement Missions,” Proceedings of the 33rd Annual INCOSE International Symposium (IS2023), Honolulu, HI, July 15-20, 2023.

Hanlon, N., and Yakimenko, O.A., “Hardening Civilian Spacecraft Against Kinetic Attack through Model-Based Systems Engineering,” Proceedings of the IEEE Aerospace Conference, Big Sky, MT, March 5-10, 2023.

Section Header

INTEGRATION WITH MATHWORKS' SYSTEM COMPOSER

Mission Architecture in System Composer



MATHWORKS' MBSE TOOLS (R2021A)

System Composer allows users to:

- Create architectures and requirements, allocate between them, and define metadata
- Create custom viewpoints of an architecture model
- Specify and simulate model behaviors using activity diagrams, sequence diagrams, Simulink, Simscape™, and Stateflow®
- Write analyses based on element properties and verify system requirements
- Import and export models using MATLAB® tables, generate reports using Simulink Report Generator™
- Organize System Composer architectural data using Simulink projects, data dictionaries, and model comparison
- Author, simulate, and deploy software architectures and generate code

Requirements Toolbox allows users to:

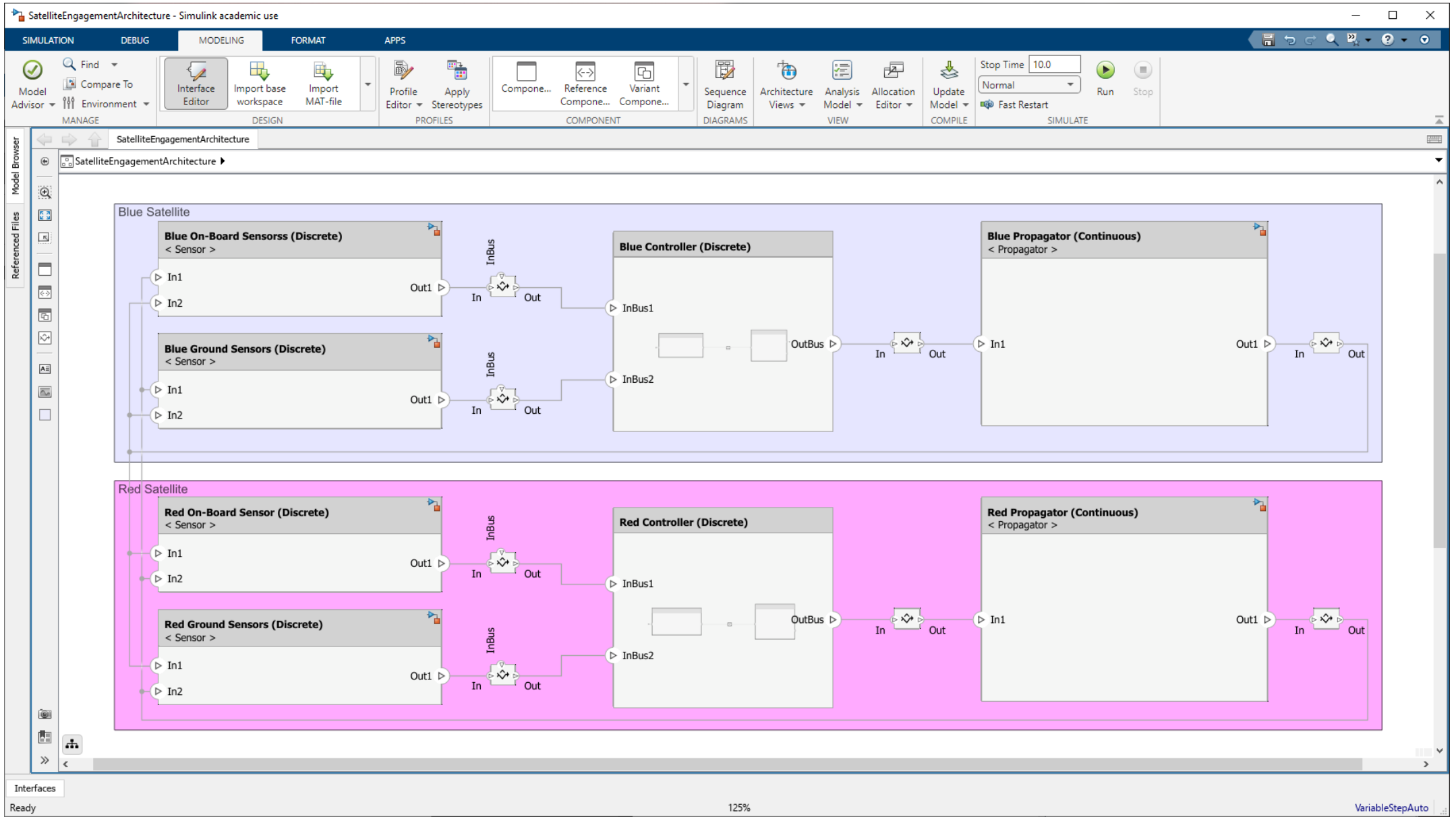
- Author requirements, import, export, and formalize requirements
- Link requirements to designs, code, and tests, specify relationships, add markup, review traceability to models and code, check consistency
- Requirements-based model verification, interpreting and reporting test results
- Requirements review and comparison, track changes and provide justification, generate reports for sign-off and approval
- Requirements traceability supported by the legacy RMI interface
- Qualify Requirements Toolbox for DO and IEC certification

- Simplified GUI
- Functions of multiple architecture models: requirements, parametric, and state machine diagram condensed into requirements editor and test manager

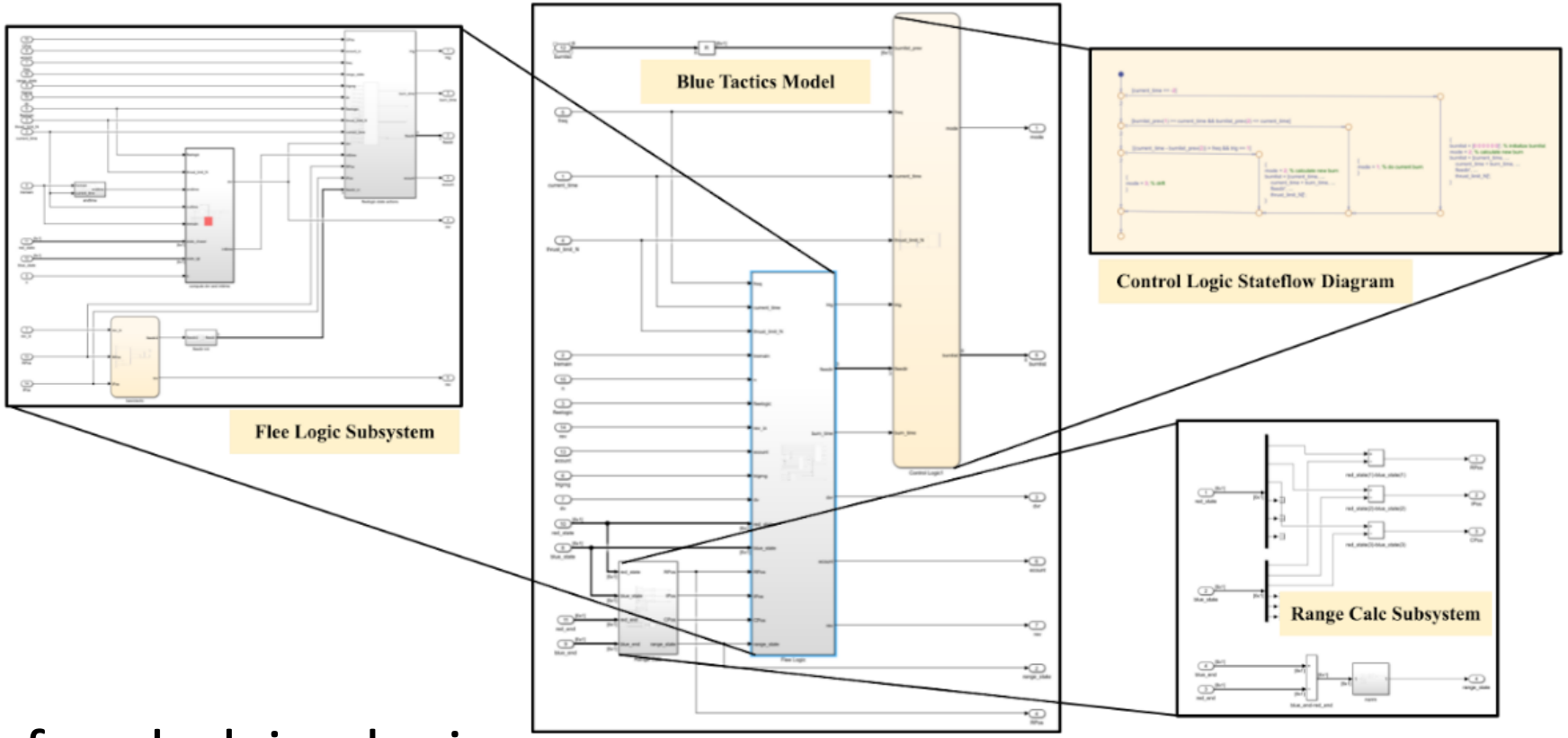
Simulink Test allows users to:

- Create and import data, develop test sequences, log signals, assess simulation and output
- Run tests, view and interpret test results, debug tests, and check test coverage
- Report and archive test specifications and test results, package test files, work with change management systems
- Create and run tests using command-line functions and scripts, set test preferences

MISSION ARCHITECTURE IN SYSTEM COMPOSER

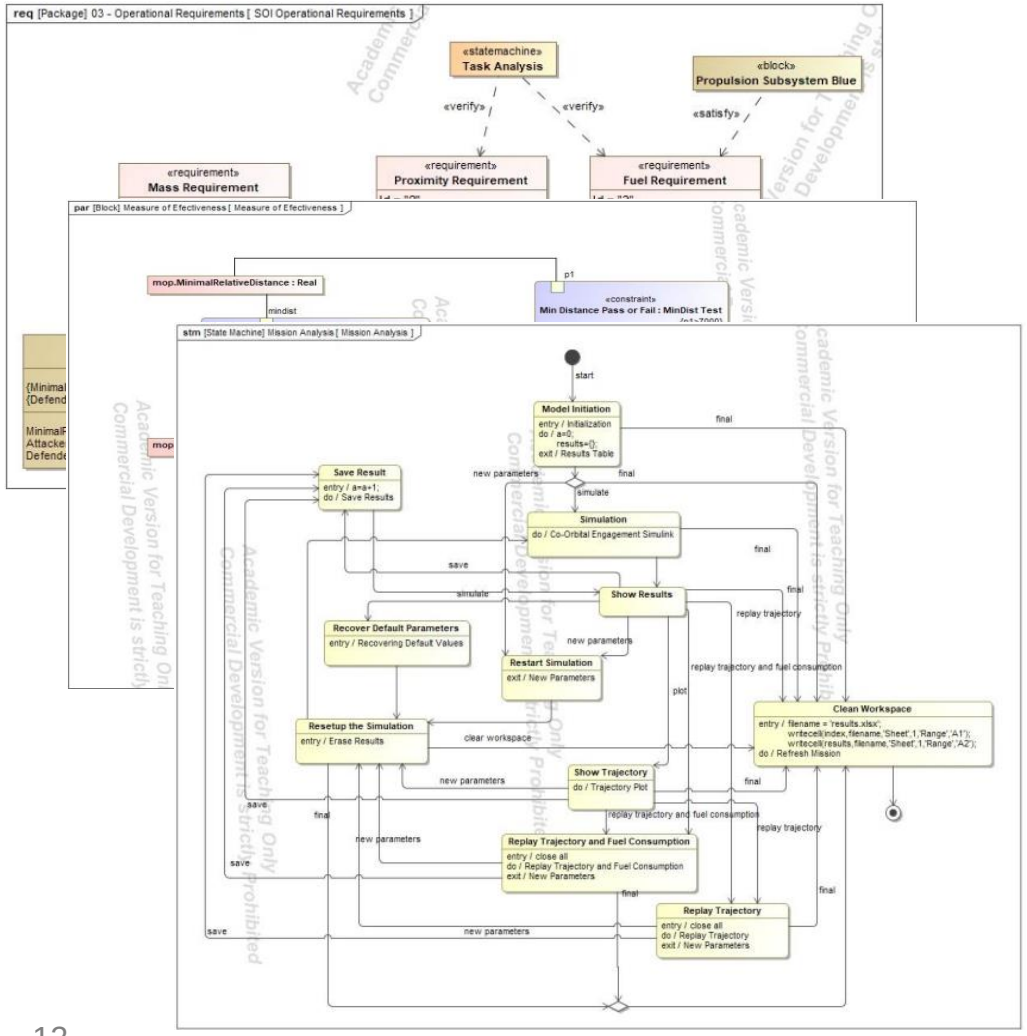


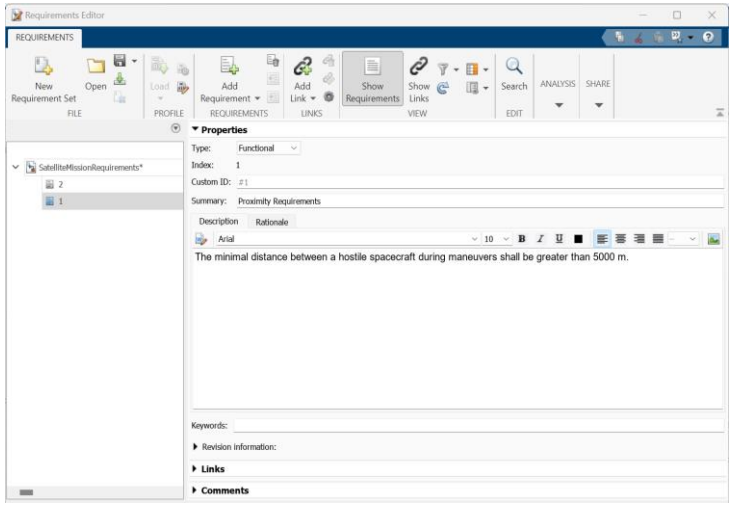
SIMULINK SUBSYSTEMS AND STATEFLOW DIAGRAMS WITHIN TACTICS COMPONENT



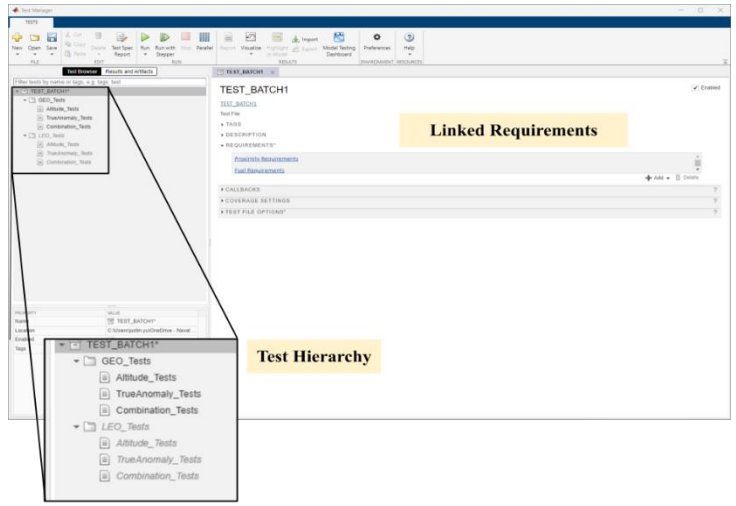
- Visual representation of underlying logic
- Simulation performance improvements
- Requirements traceability and verification
- Increased modularity to adapt to changing requirements and model updates

SIMPLIFICATION OF REQUIREMENTS AND ASSESSMENTS GUI





Requirements Editor

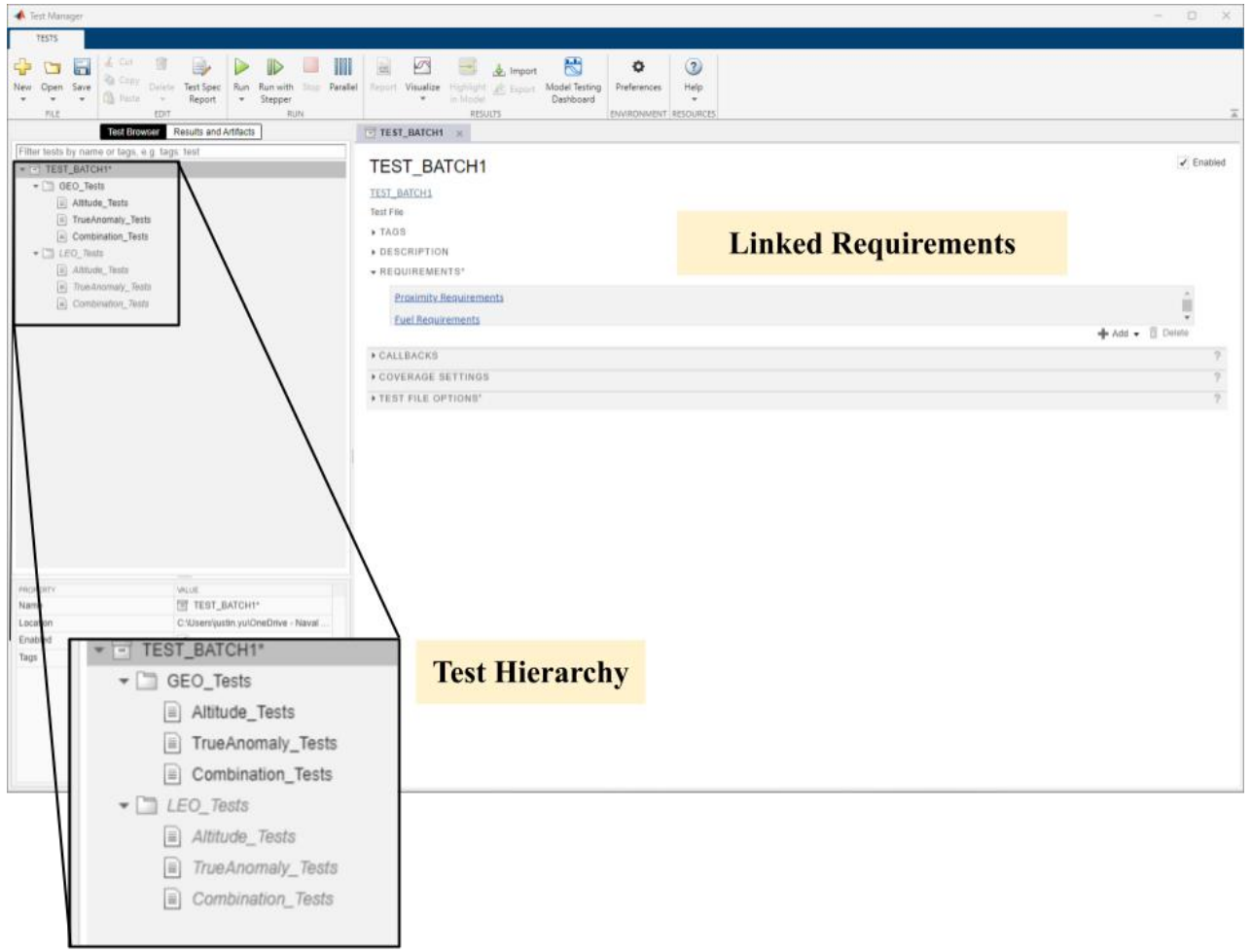


Test Manager

Section Header

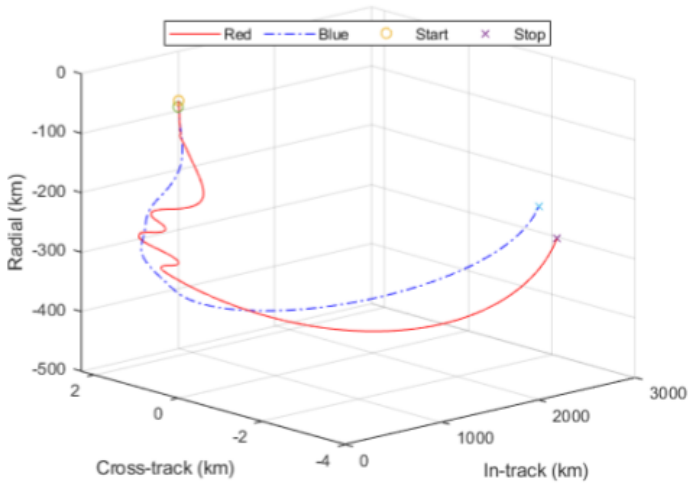
DEMONSTRATION OF MODEL FEATURES/EXECUTION

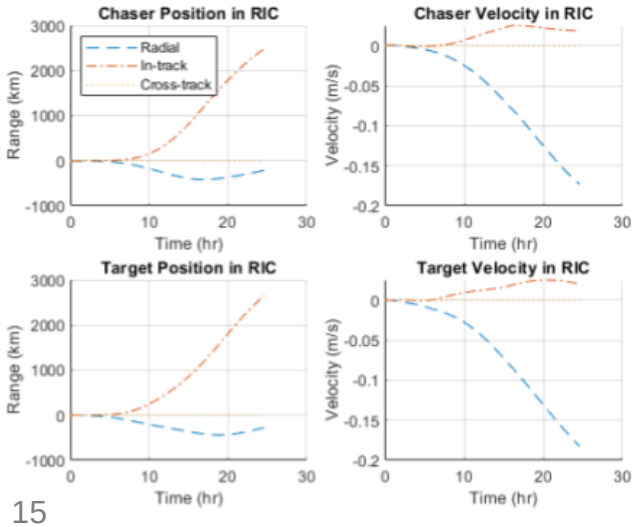
TEST MANAGER GUI FOR MANAGING BATCH TESTS

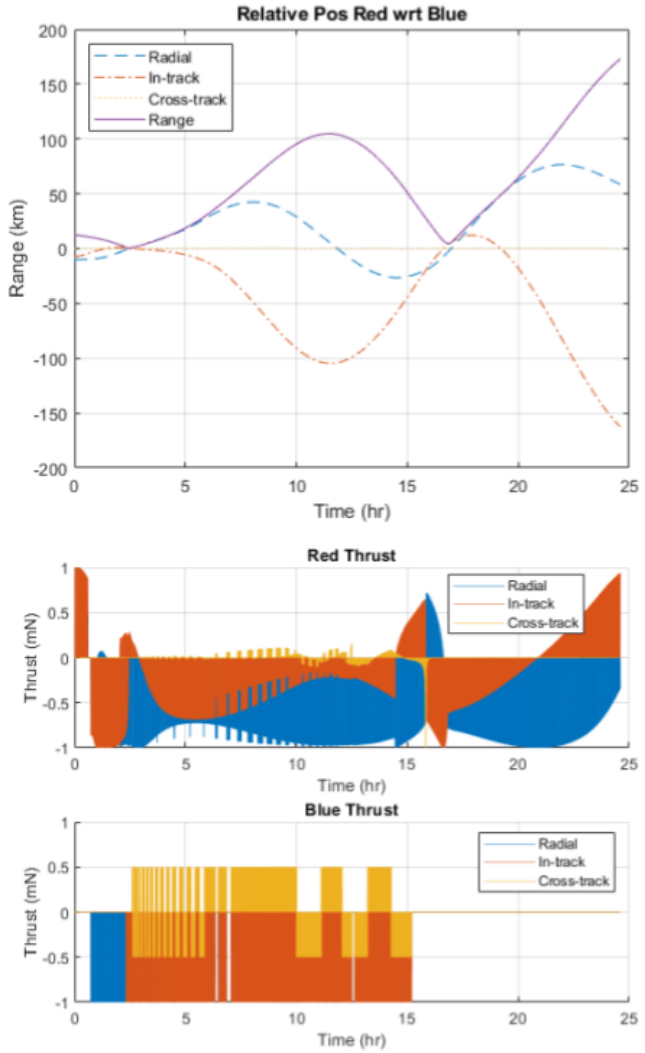


IN-DEPTH REQUIREMENT VERIFICATION

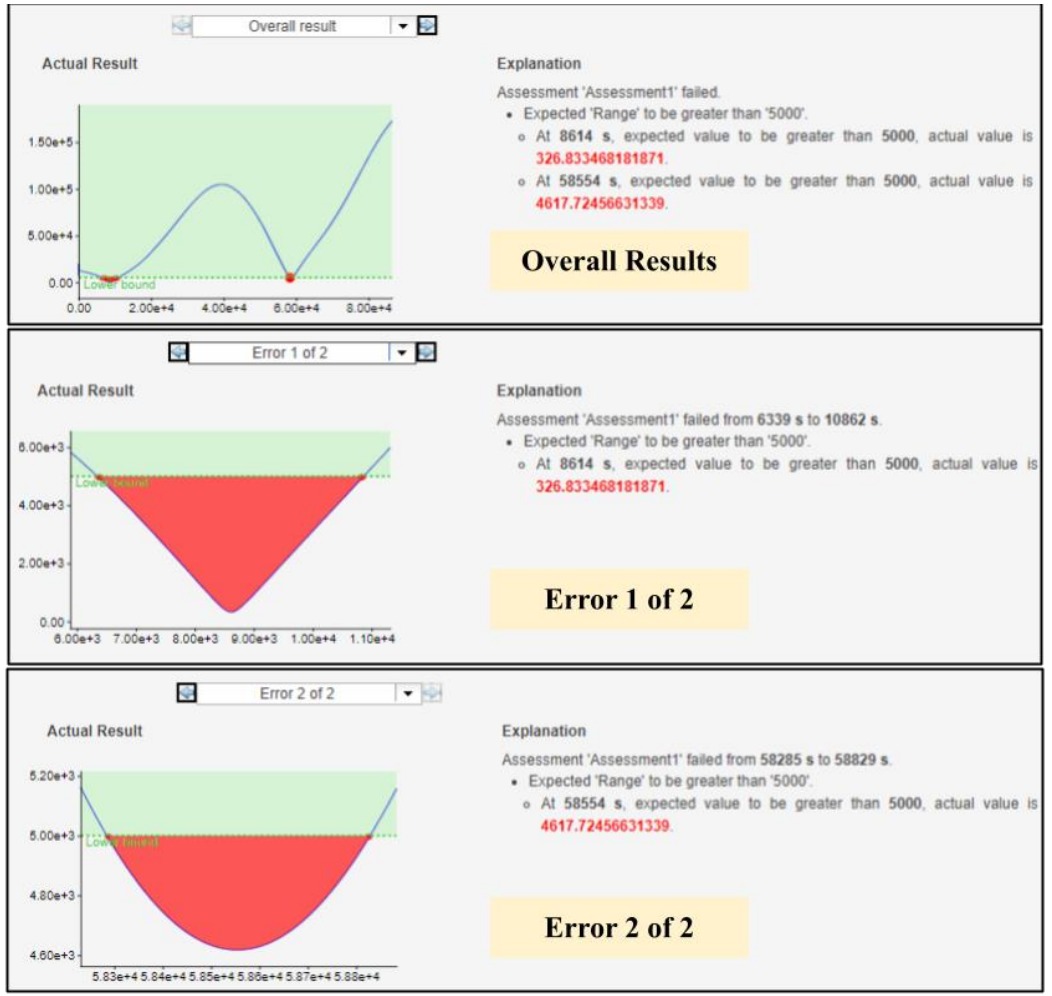
GEO chaser trajectory starting 10 km below and 0.01° true anomaly behind target







Automatic assessment conducted by Test Manager (proximity requirement)



Section Header

CONCLUSIONS

CONCLUSIONS

- Successful integration
- Simplified GUI; batch-testing; single source control tool; no value compatibility issues
- Complexity of translating MATLAB into Simulink/Stateflow
- Requirements reader still under construction
- Does not allow not physical signals – can’t simulate any system