



INCOSE 2026 August SysML2 Exchange Executing SysML2 for Operational Systems

Richard Page

Ansys Digital Mission Engineering MBSE Technical Lead

2026-08-26

©2026 ANSYS, Inc.

Personal Background: Richard Page

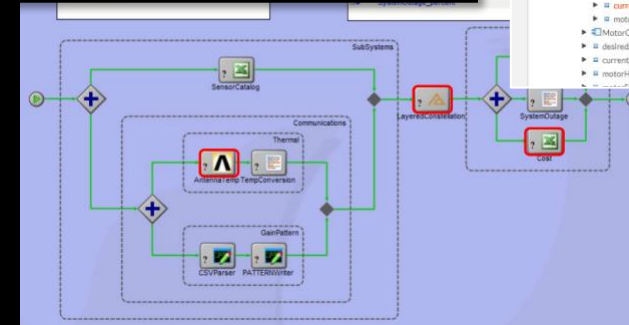


- B.S. Aerospace Engineering University of Illinois Urbana-Champaign
- M.S. Aerospace & Computational Science Engineering UIUC
 - Computational Mechanics (Finite Element)
 - Computational Astrodynamics and State-Space Simulation
 - 3D graphics (CSG, Tensor/Quaternion representation, ...)
- Graduate Intern at NASA Goddard Spaceflight Center
- AGI R&D for Dev Kit APIs, Enterprise Data, STK, ODTK, SSA, and now MBSE
- Ansys Government Initiatives / Digital Mission Engineering Business Unit
- **2018-Present: Principal Technical Lead for DME MBSE and primary SysML2 expert at Ansys, now Synopsys**
- **2023-Present: SysML Execution WG chair**

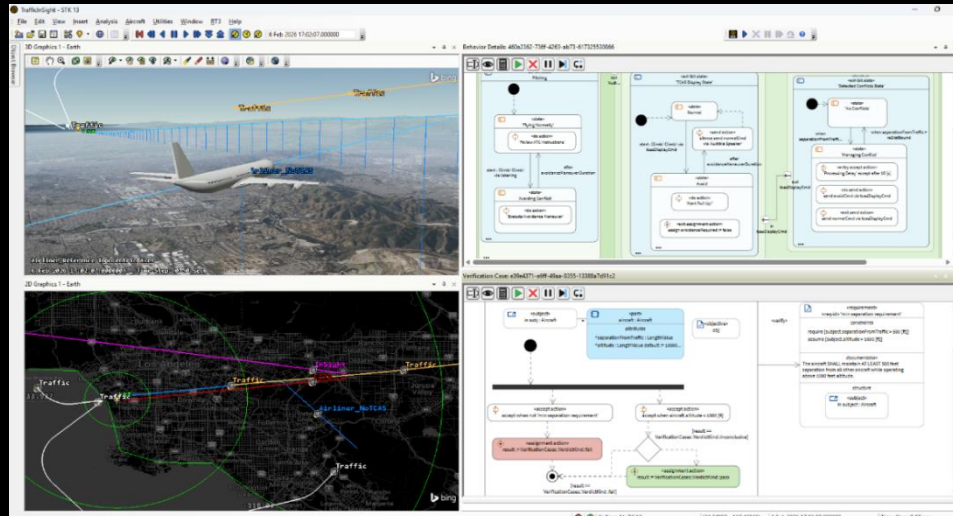


Overview

- Tool Overview from Ansys, part of Synopsys
 - SysML2 Modeler
 - Software Systems Engineering
 - Requirement V&V Workflow
- SysML v2 Execution Demonstration
 - Operational Requirements Verification Cases
 - Executing Dynamic Systems in SysML



Showcase
Execution
Demo!

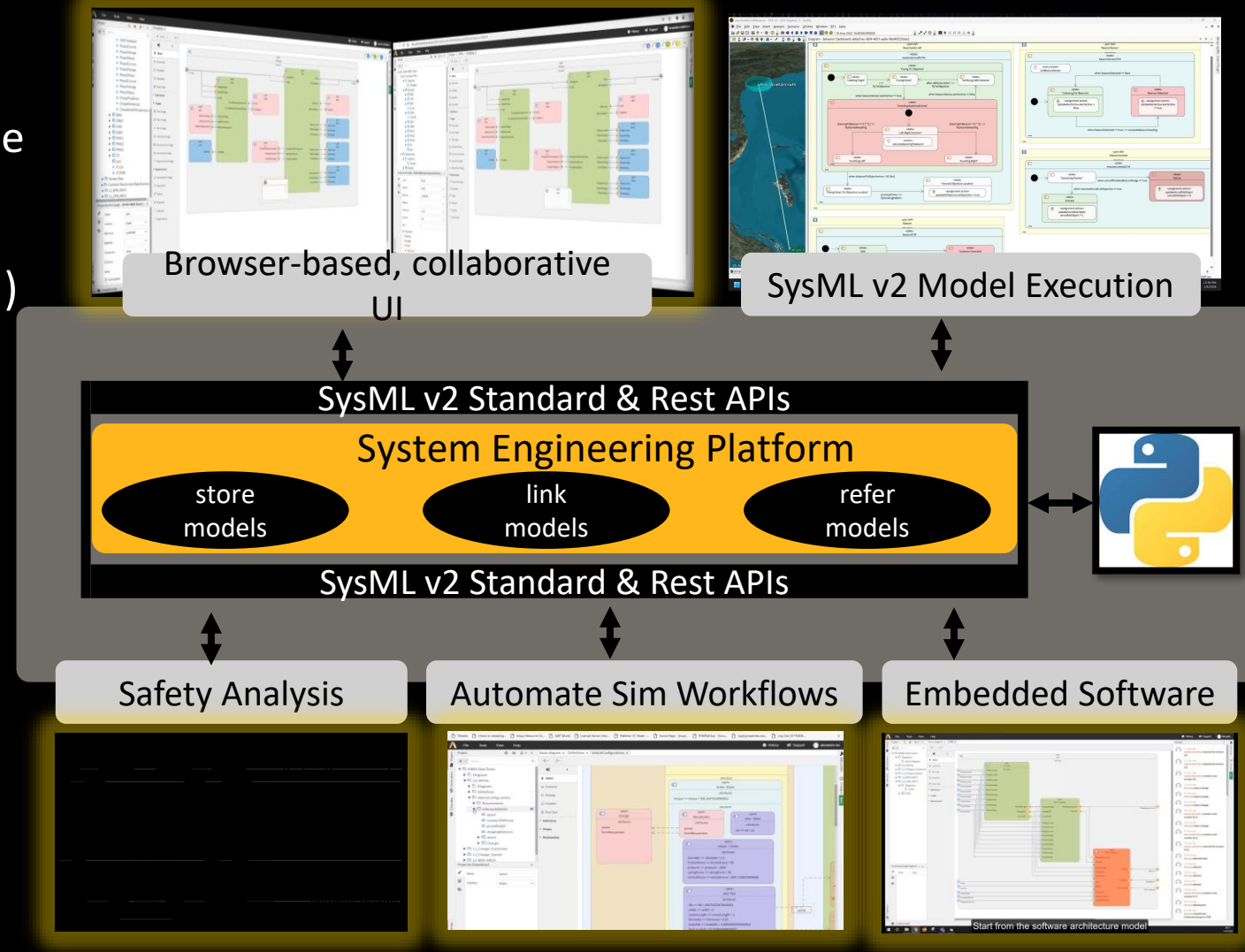




Connecting to SysML 2 Ansys Tools

Digital Engineering / Ansys MBSE Solutions Overview

- Systems Engineering Platform - **real-time collaboration**, federated with other Enterprise Systems, **standardized remote APIs**
- **SysML v2** System Architecture Modeler (SAM) - **cloud-native, thin UI**
- Integrated **Engineering workflows**: Verification/Optimization, Safety, Cyber security, Embedded SW
- Adaptable **Digital Engineering methodology** delivered as SysML v2 library
- Python APIs for **Interoperability**
- SysML v2 **Model Execution**



Ansys System Architecture Modeler (4 min lightning intro)

<https://www.ansys.com/products/connect/ansys-system-architecture-modeler>

Ansys System Architecture Modeler (SAM): Web-based modeler

SSO

Models management

Collaborators management

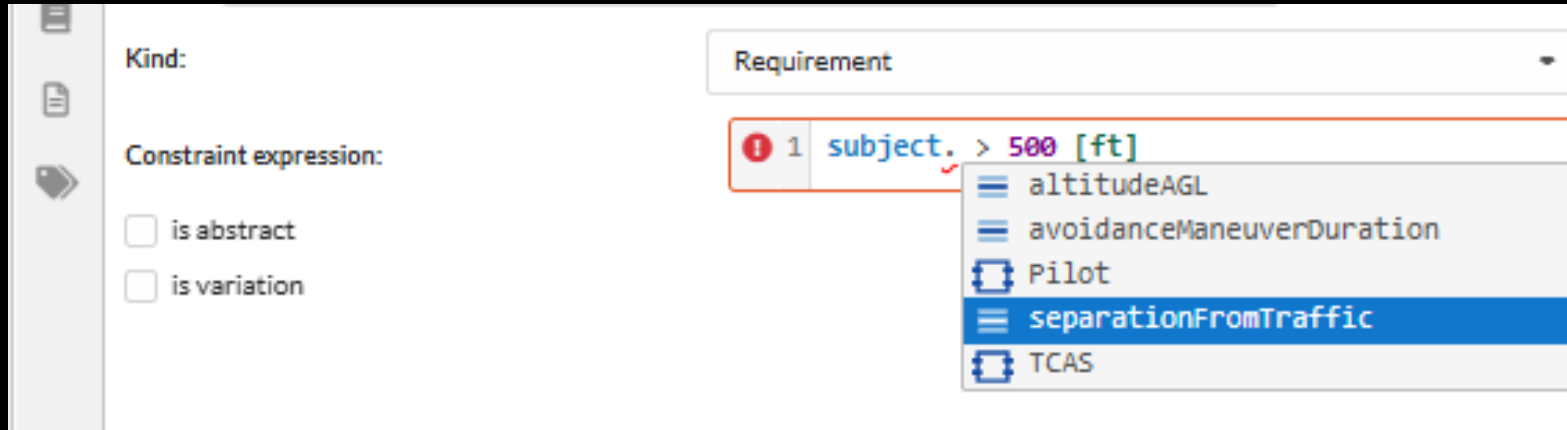
Live Collaboration

Textual & SysML API Support

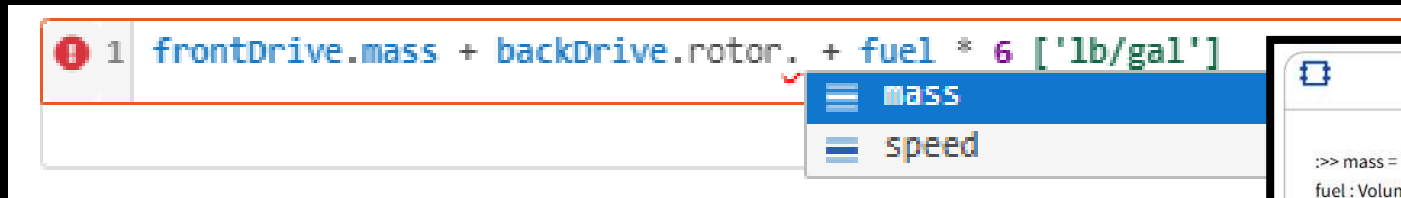
Version management

SysML v2 modeler

SAM: SysML v2 Expression editing

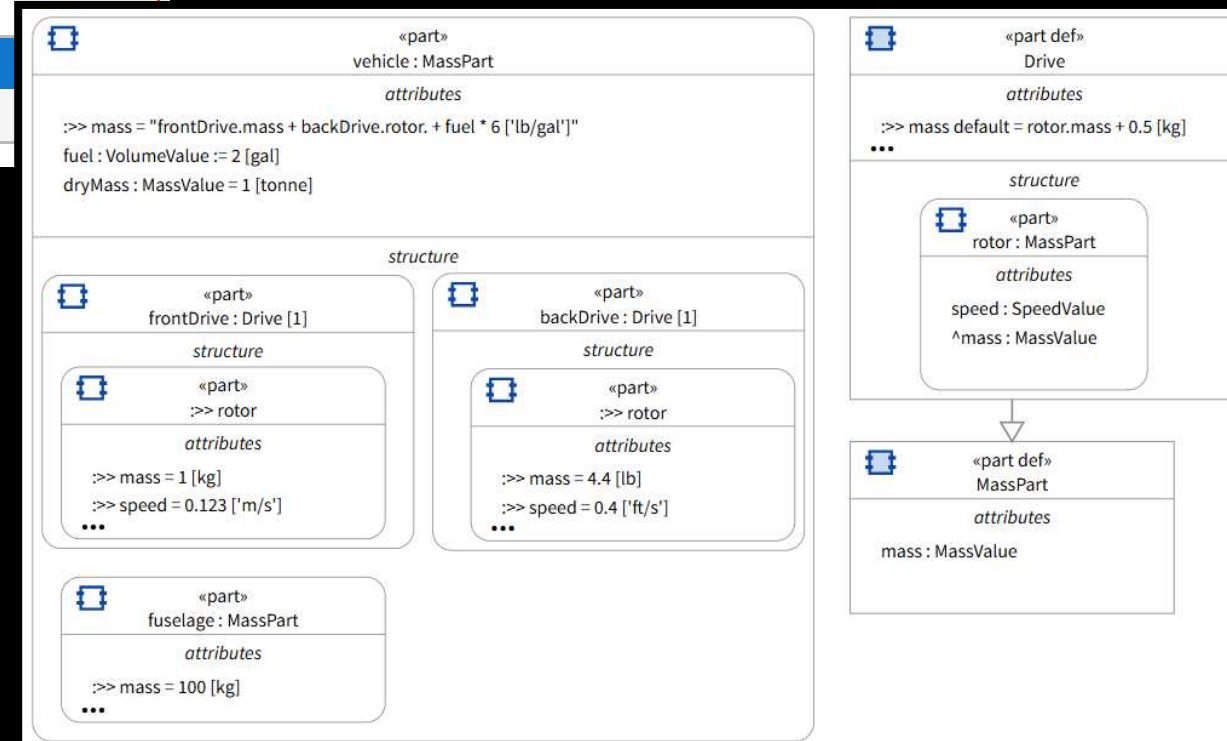


Expressions both evaluated at “System Level” And “Instance Level” during execution



Expression Auto-Completion
With Full Dimension & Units Support

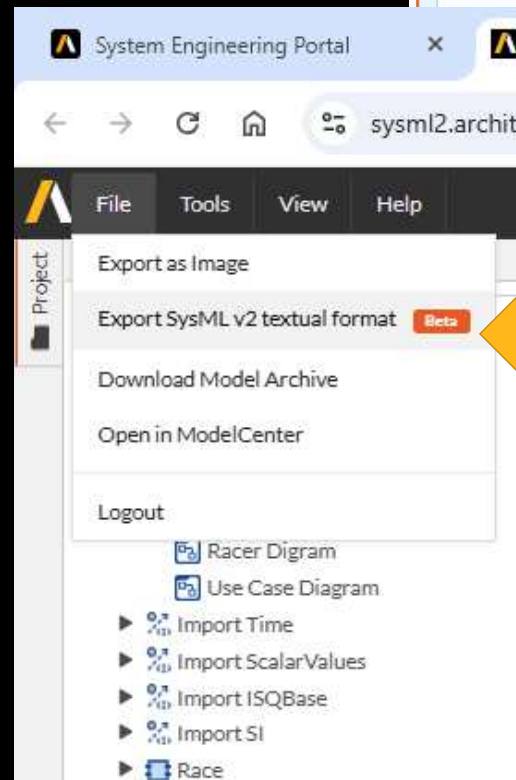
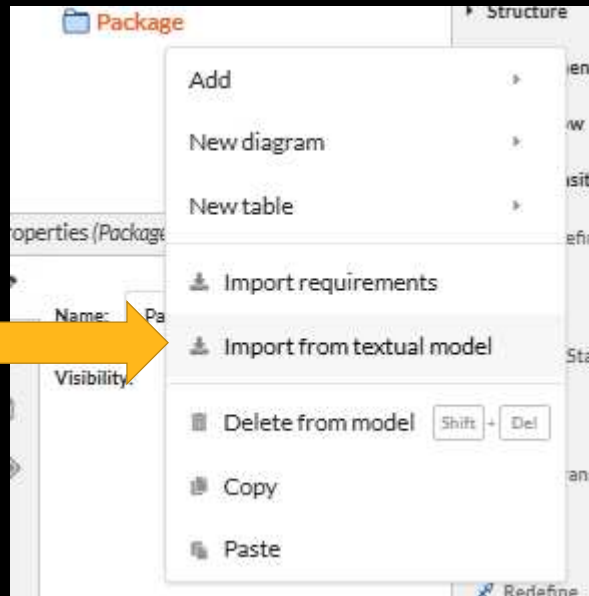
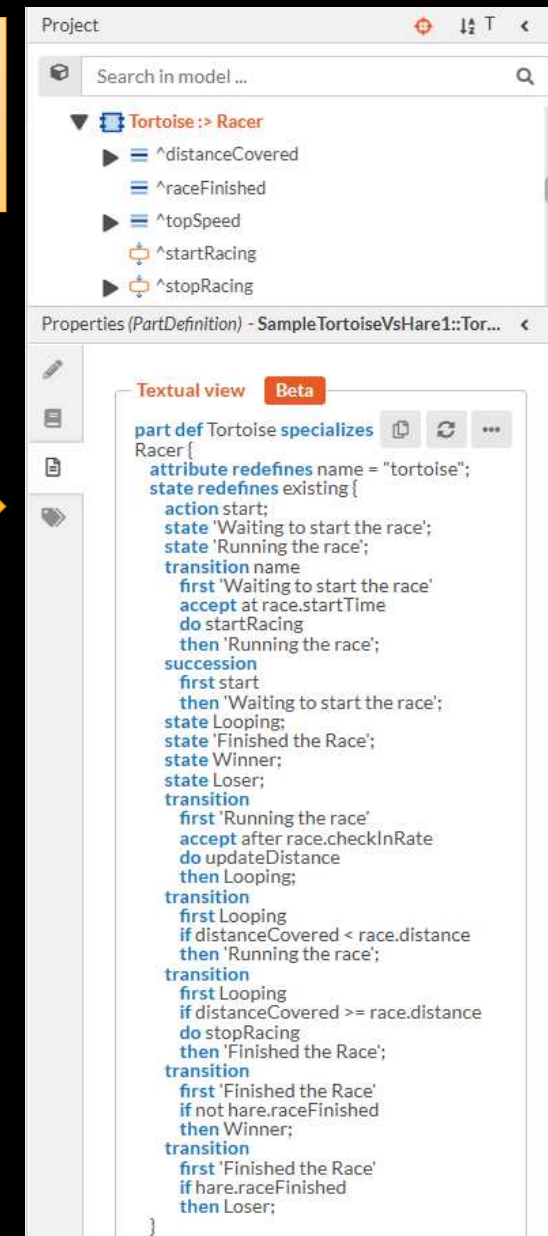
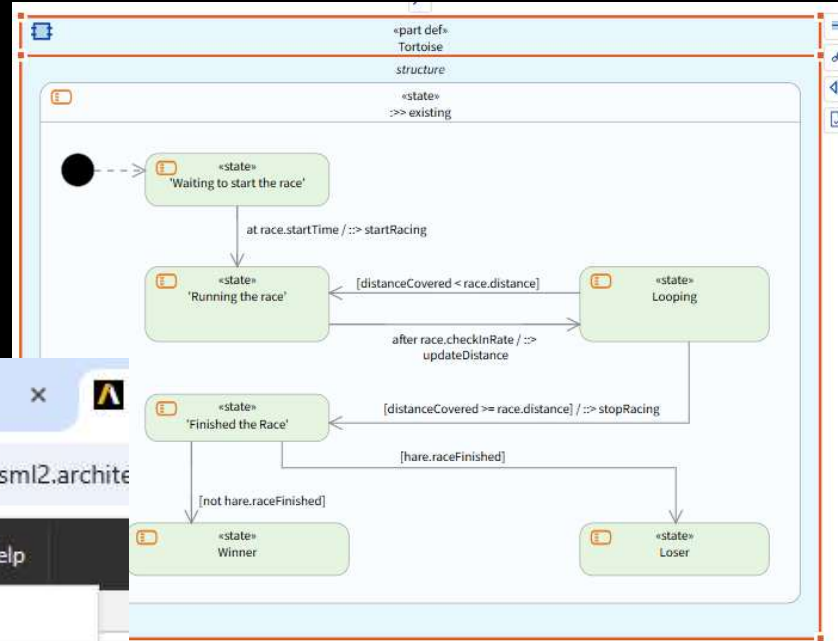
Unit conversions performed during execution



New: SysML textual import & export

Currently Released as
Beta Capability, pending full
tool support for SysML 2.0.0

- SAM 2026 R1
 - Import SysML v2 textual file
 - Export SysML v2 textual format from any model element



Export *.sysml to file
or view in editor

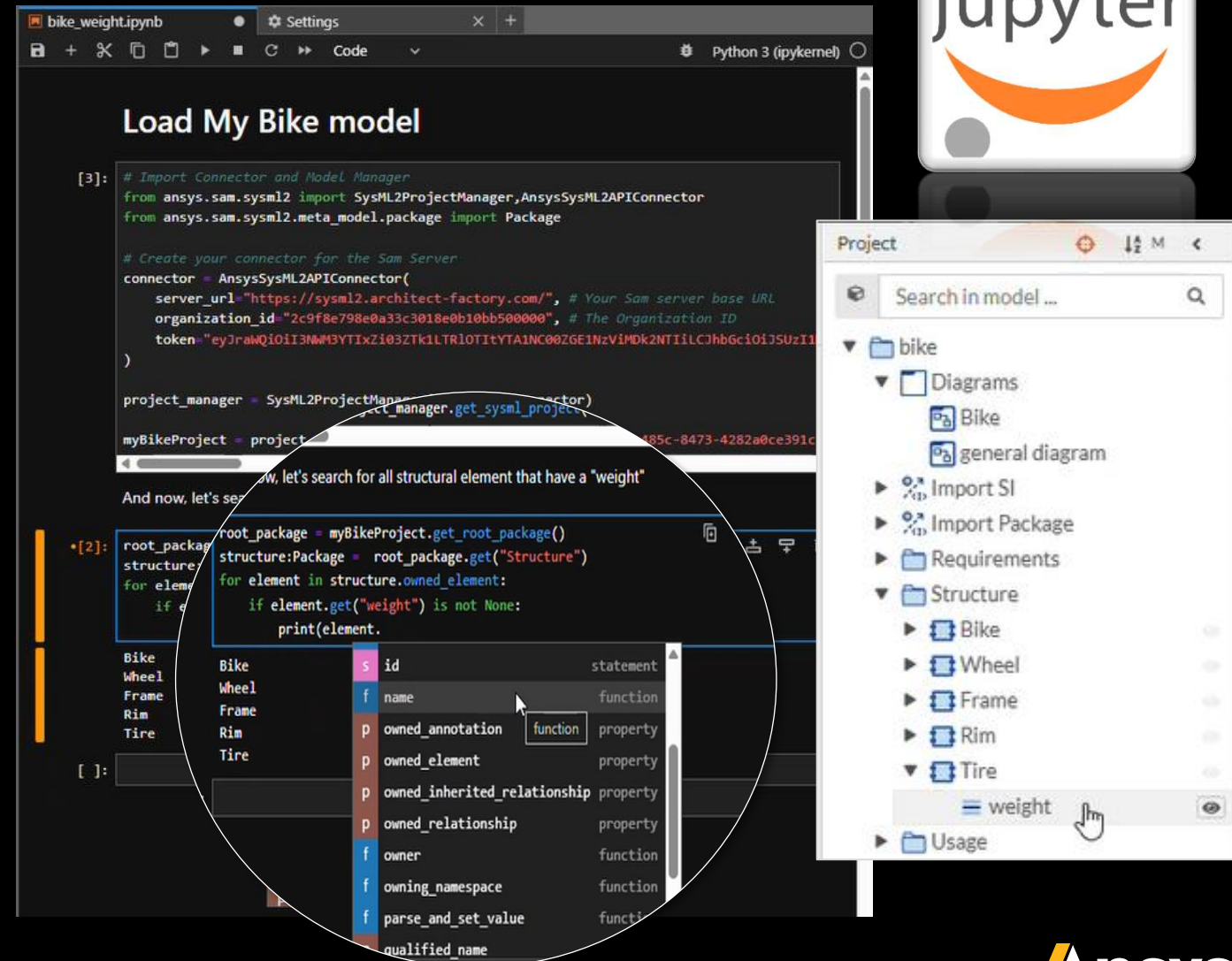
Import *.sysml
to package

Model API / PySAM: Python Library

Model *Automation* and Interop



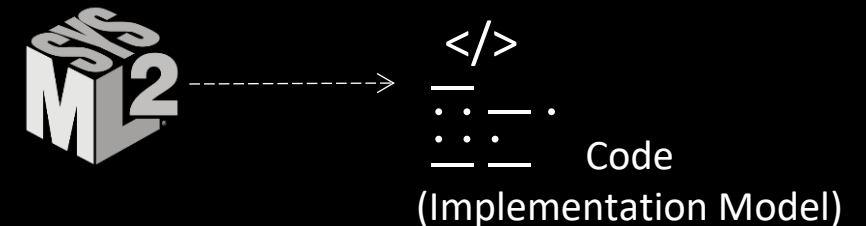
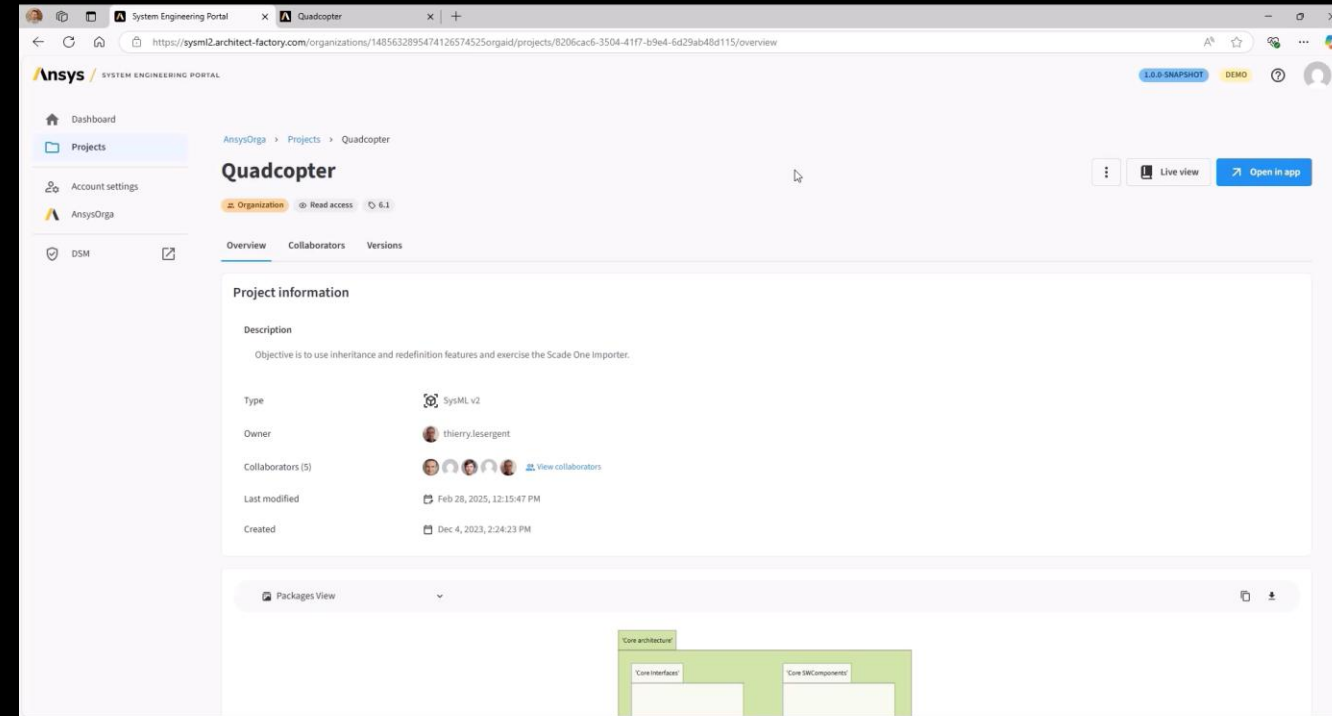
- Access to SAM REST API from Python
 - Examples and documentation
- Python library extended with static Python classes
 - Allows for accurate autocompletion in any code IDE
- Transactional mode
 - Higher writing performance
- **PySAM available as OpenSource library from PyAnsys:**
 - <https://sysml2.docs.pyansys.com/version/stable/>

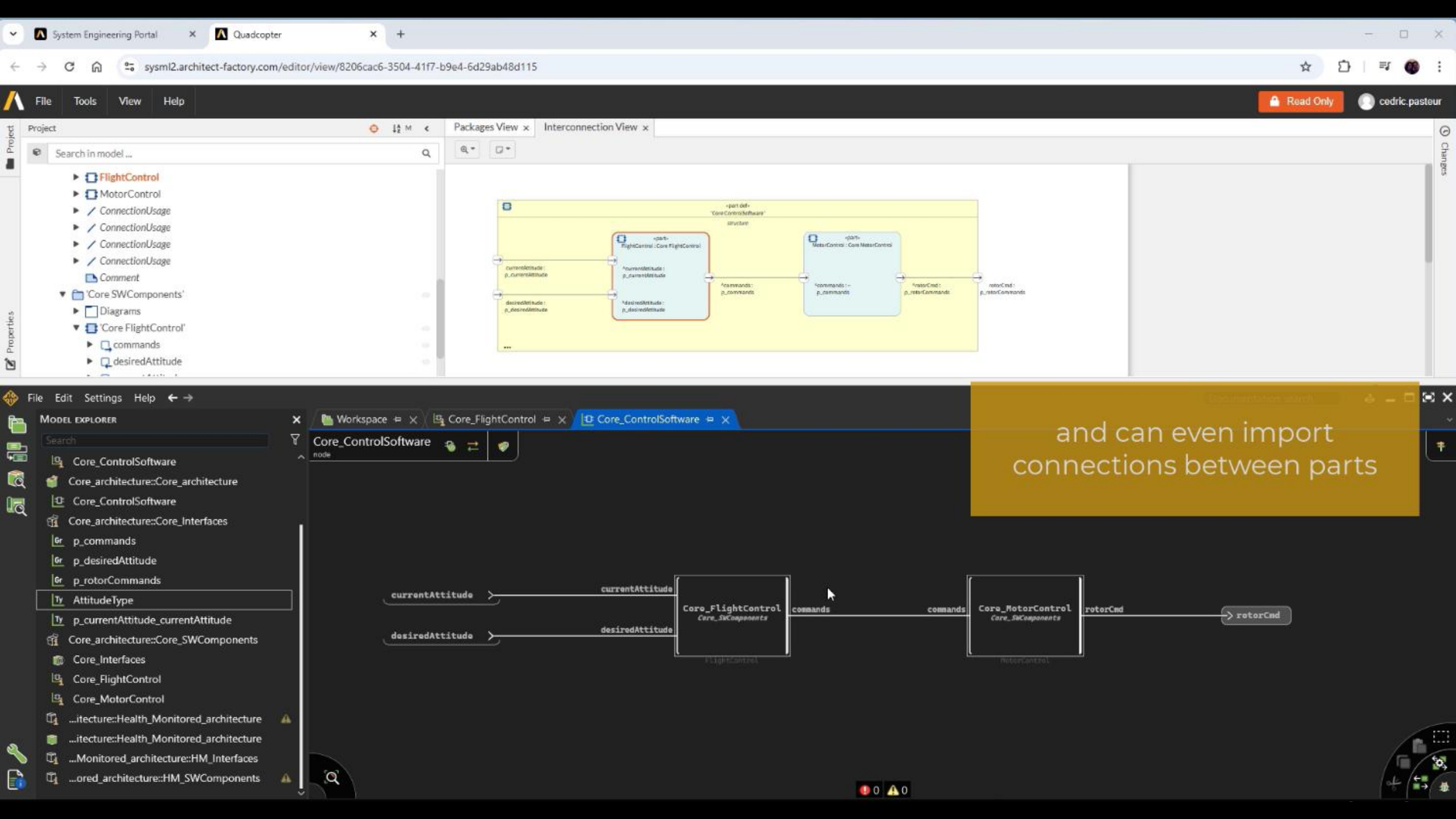


Digital Engineering / Software System Engineering Workflows

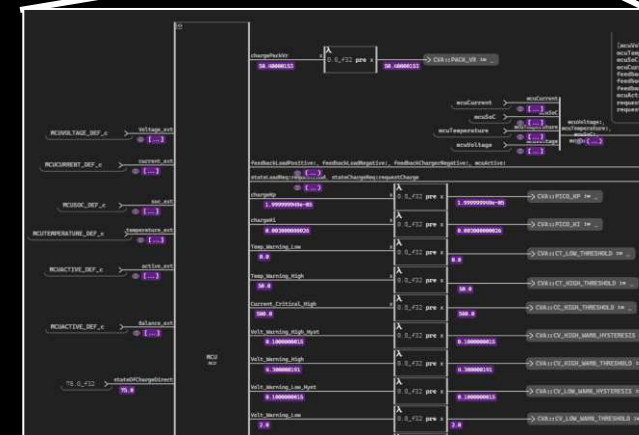
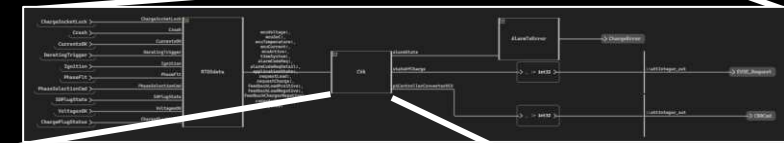
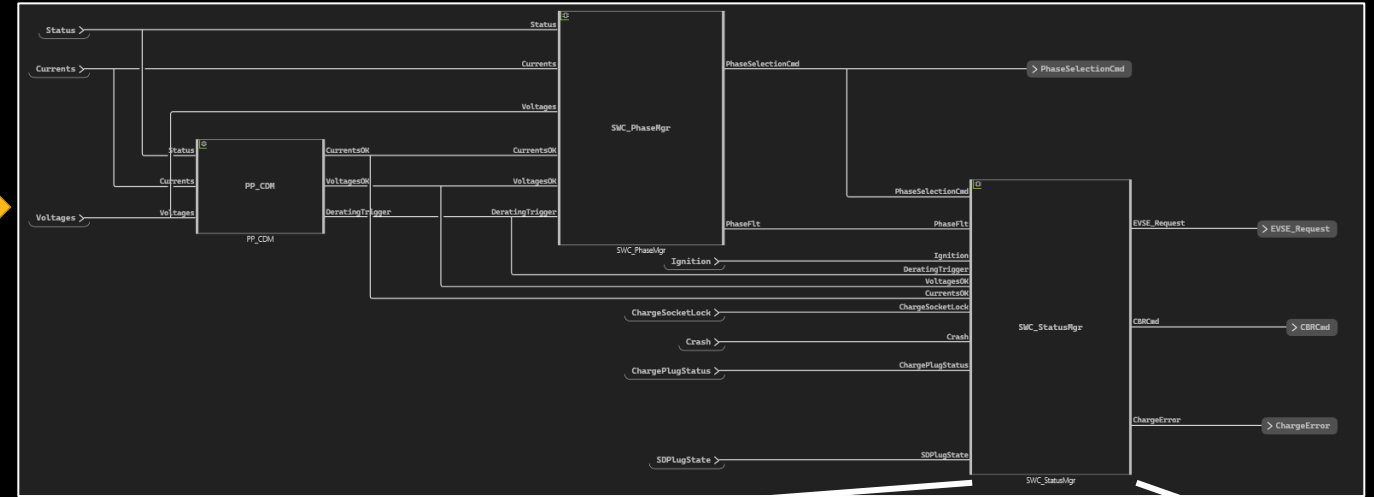
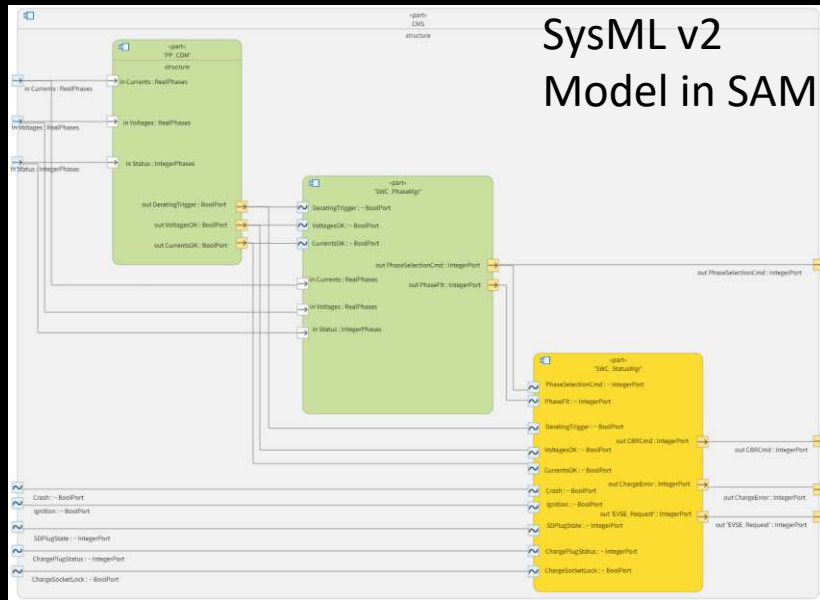
Ansys SAM supports engineering workflows

- from system architectural models to software architectural models
- connects to a dedicated **embedded software modeling language** for software prototyping and detailed software design
- **certified code generation** for mission and safety-critical systems





Connected to Embedded Software development with Scade One

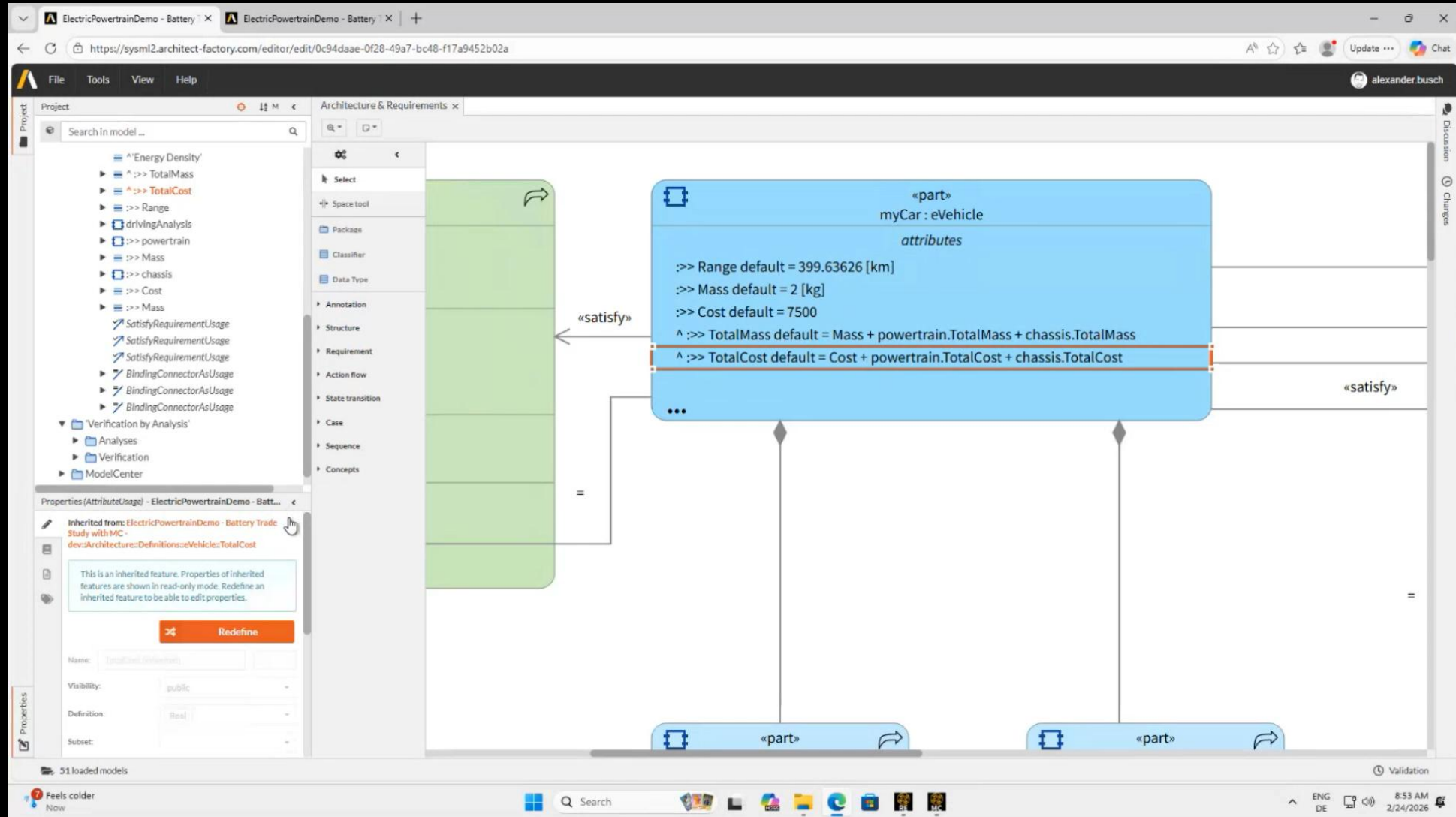


Process:

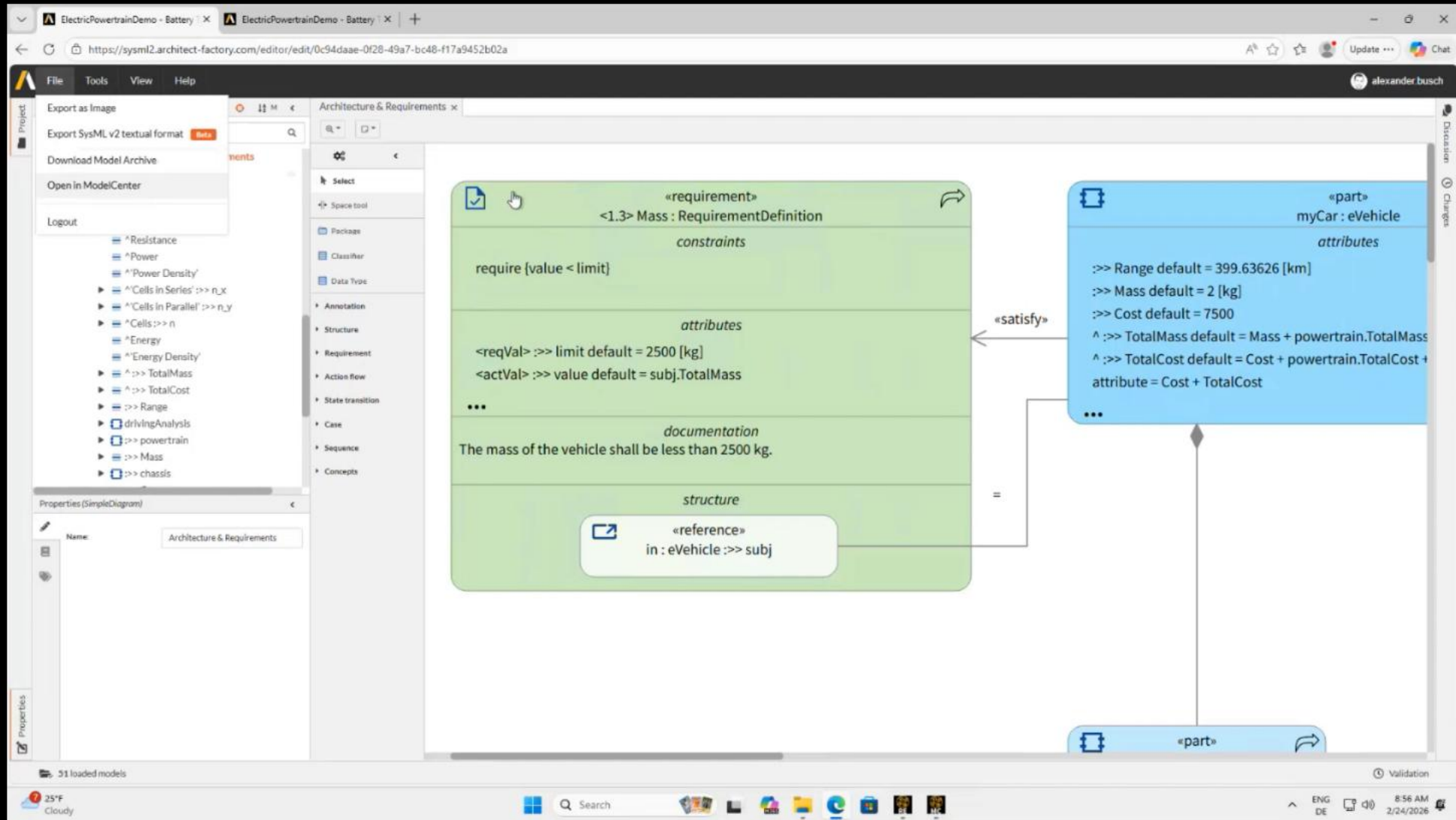
- Collaborative SW Architecture modeling in SAM
- SAM model import in Scade One
- Detailed embedded SW design
- Simulation / Design debug
- Code generation with qualified code generator
- Testing (incl. model coverage)

Embedded SW
detailed design,
here in debug mode

Ansys ModelCenter MBSE – Workflows and Requirement V&V (4min)



Ansys ModelCenter MBSE – Workflows and Requirement V&V



Ansys ModelCenter MBSE – Workflows and Requirement V&V

The screenshot displays the Ansys ModelCenter MBSE interface, showing a project titled "ElectricPowertrainDemo - Battery". The main workspace displays two models: a requirement model (green box) and a part model (blue box).

Requirement Model (Left):

- «requirement»
- <1.3> Mass : RequirementDefinition
- constraints
- require {value < limit}
- attributes
- <reqVal> :>> limit default = 2500 [kg]
- <actVal> :>> value default = subj.TotalMass

Part Model (Right):

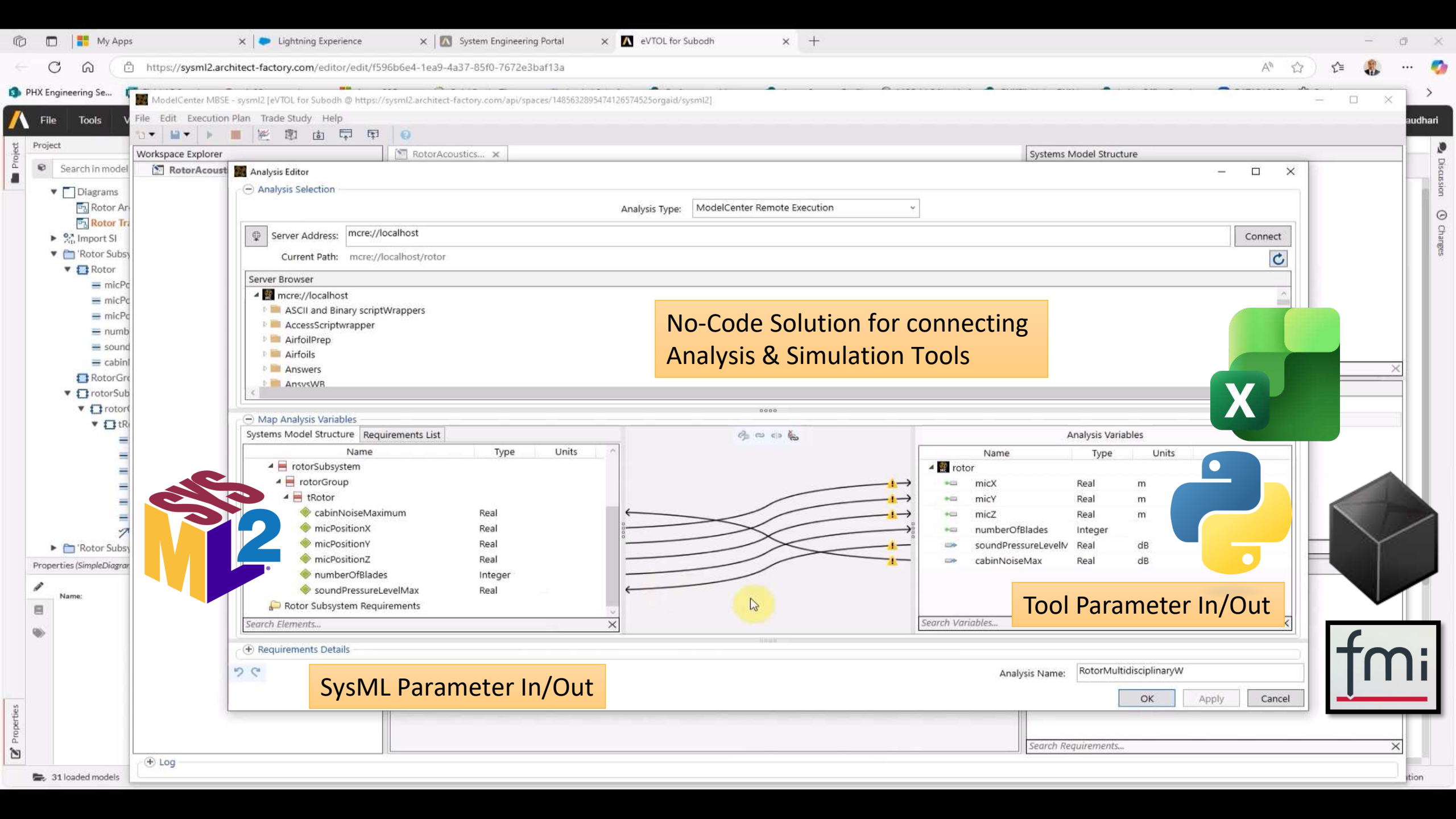
- «part»
- myCar : eVehicle
- attributes
- >>> Range default = 399.63626 [km]
- >>> Mass default = 2 [kg]
- >>> Cost default = 7500
- ^>>> TotalMass default = Mass + powertrain.TotalMass
- ^>>> TotalCost default = Cost + powertrain.TotalCost + attribute = Cost + TotalCost

A "«satisfy»" relationship is indicated between the requirement and the part model.

The interface also includes a Project Explorer on the left, a Properties (SimpleDiagram) panel at the bottom left, and a Workspace Explorer at the bottom center. The Workspace Explorer shows a tree structure of models, including "1 - Mass + Cost Roll-up", "2 - Early-On Battery Sizing & Range Est", "3 - eVehicle Range + Mass + Cost V", and "Execution Plan 2*". The bottom right panel displays the Systems Model Structure, including a table of Structure Elements and a list of Analyses.

Name	Value	Type	Units	Associated Analyses
ElectricPowertrainD				
myCar Usage				
myCar				
Cost	7500.0	Real		{1.4 vehicle.Cost} <myC

Name	Associated Analyses
ElectricPowertrainDemo - Battery Trade	
Requirements	
1 vehicle	
1.3 Mass	{1.3 vehicle.Mass}



No-Code Solution for connecting
Analysis & Simulation Tools

Tool Parameter In/Out

SysML Parameter In/Out

Requirement Verification Results Saved back to SAM Model

The screenshot displays the SAM Model software interface. On the left is a project tree with folders like 'Range + Mass + Cost Verification', 'Diagrams', and 'ModelCenter Requirements'. The main area shows a table of verification results. The table has columns: Name, ExecutionPlan, ExecutionPlanResult, ReqID, Text, and Satisfied. It lists several requirements (ReqID 1.3, 1.2, 1.4) and their verification status (Satisfied). Below the table are buttons for 'Add to table context', 'Add column', and 'Delete'. The bottom of the interface shows a 'Properties' panel with the message 'No properties for the current selection'.

Project

File Tools View Help

Architecture & Requirements x ModelCenter Requirements Verification Table x ModelCenter Requirements Verification Table x ModelCenter Requirements Verification Table x

Search in model ...

Table context: Range + Mass + Cost Verification - Satisfied Expression: @Package or @Annotation Hide Inherited:

Name	ExecutionPlan	ExecutionPlanResult	ReqID	Text	Satisfied
1.3	"3 - eVehicle Range + Mass + Cost Verification"	"Range + Mass + Cost Verification - Satisfied"	1.3	The mass of the vehicle shall be less than 2500 kg.	true
1.2	"3 - eVehicle Range + Mass + Cost Verification"	"Range + Mass + Cost Verification - Satisfied"	1.2	The range of the vehicle with a battery SoC = 1.0 shall be equal or larger than 390 km for the "San Francisco" drive cycle scenario.	true
1.4	"3 - eVehicle Range + Mass + Cost Verification"	"Range + Mass + Cost Verification - Satisfied"	1.4	The cost of the vehicle shall be less than 17000 \$.	true
1.2	"3 - eVehicle Range + Mass + Cost Verification"	"Range + Mass + Cost Verification - Satisfied"	1.2	The range of the vehicle with a battery SoC = 1.0 shall be equal or larger than 390 km for the "San Francisco" drive cycle scenario.	true
1.4	"3 - eVehicle Range + Mass + Cost Verification"	"Range + Mass + Cost Verification - Satisfied"	1.4	The cost of the vehicle shall be less than 17000 \$.	true
1.3	"3 - eVehicle Range + Mass + Cost Verification"	"Range + Mass + Cost Verification - Satisfied"	1.3	The mass of the vehicle shall be less than 2500 kg.	true

Add to table context Add column Delete

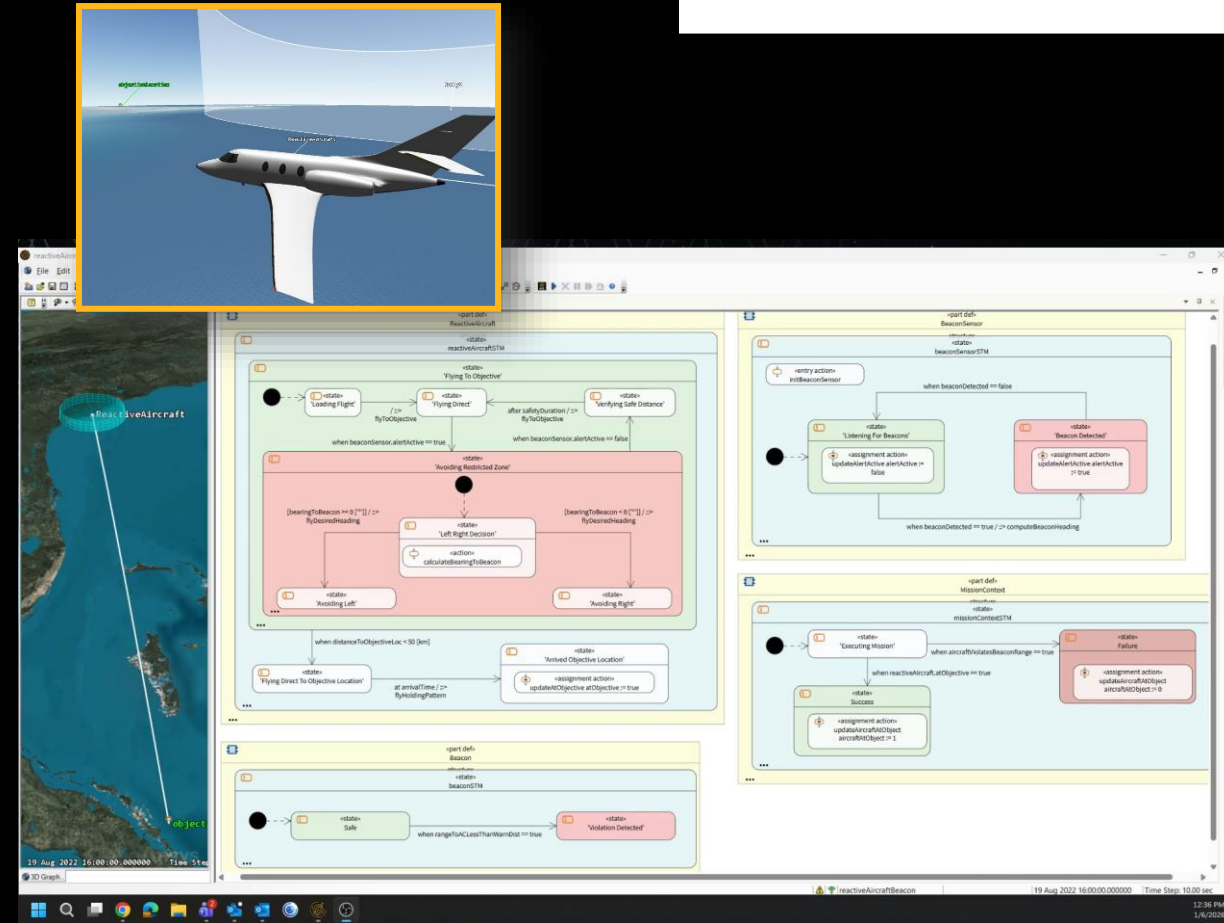
Properties

No properties for the current selection

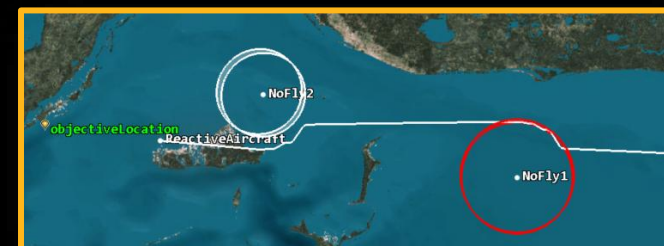
SysML v2 Model Execution (Alpha) Enhancements

Engineers and analysts can now launch SysML v2 models from Ansys SAM directly inside of Ansys STK. You can open views of existing diagrams from Ansys SAM and view animations on these diagrams during model execution. Diagrams can be used interactively to set breakpoints and associated Play, Pause, Continue, and Step controls provide robust debugging capabilities.

- STK plugin for SysML v2 model execution
- Open views of Ansys SAM diagrams
- Behaviors on diagrams animate during execution
- Interact with diagram to set breakpoints
- Play, Pause, Continue, and Step execution controls for debugging



Initial capability available **now** to early adopters!





SysML Execution Tool Demonstration

Ansys, part of Synopsys

SAM+STK+BEE : Alpha Demo

Continuous Agile Development
In coordination with System Architecture Modeler team
Following SysML v2.x

- **System Architecture Modeler**

- SysML v2 Model Editing
- Graphical Diagrams

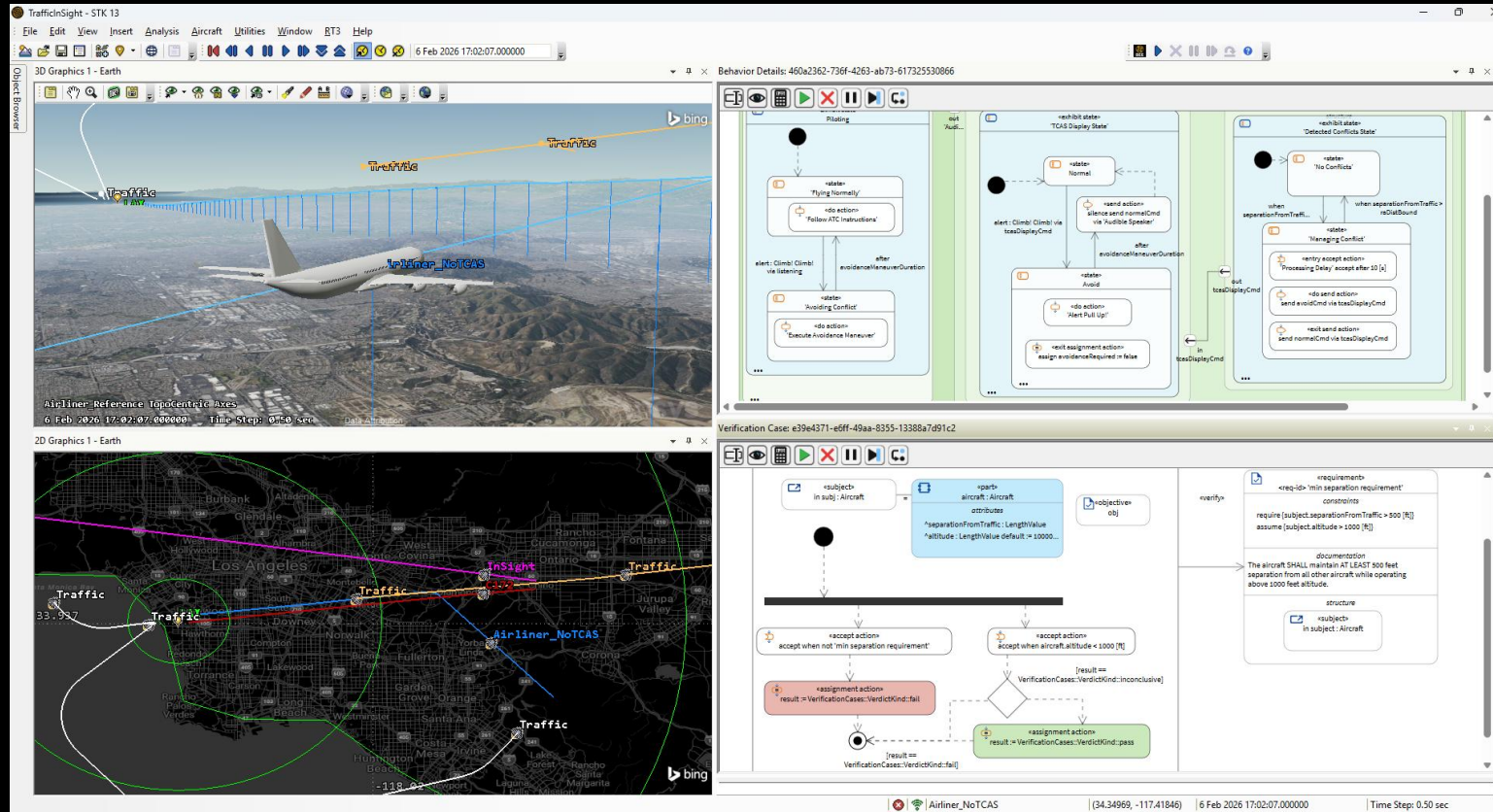
Demo Shown uses a development build of 2027r1
New capability available second half 2026

- **Systems Tool Kit (STK) ®**

- 4D geospatial modeling
- Event detection
- Reporting & Graphing

- **Behavior Execution Engine**

- SysML Execution
- Breakpoints and Live Evaluation
- Distinguishing Instances

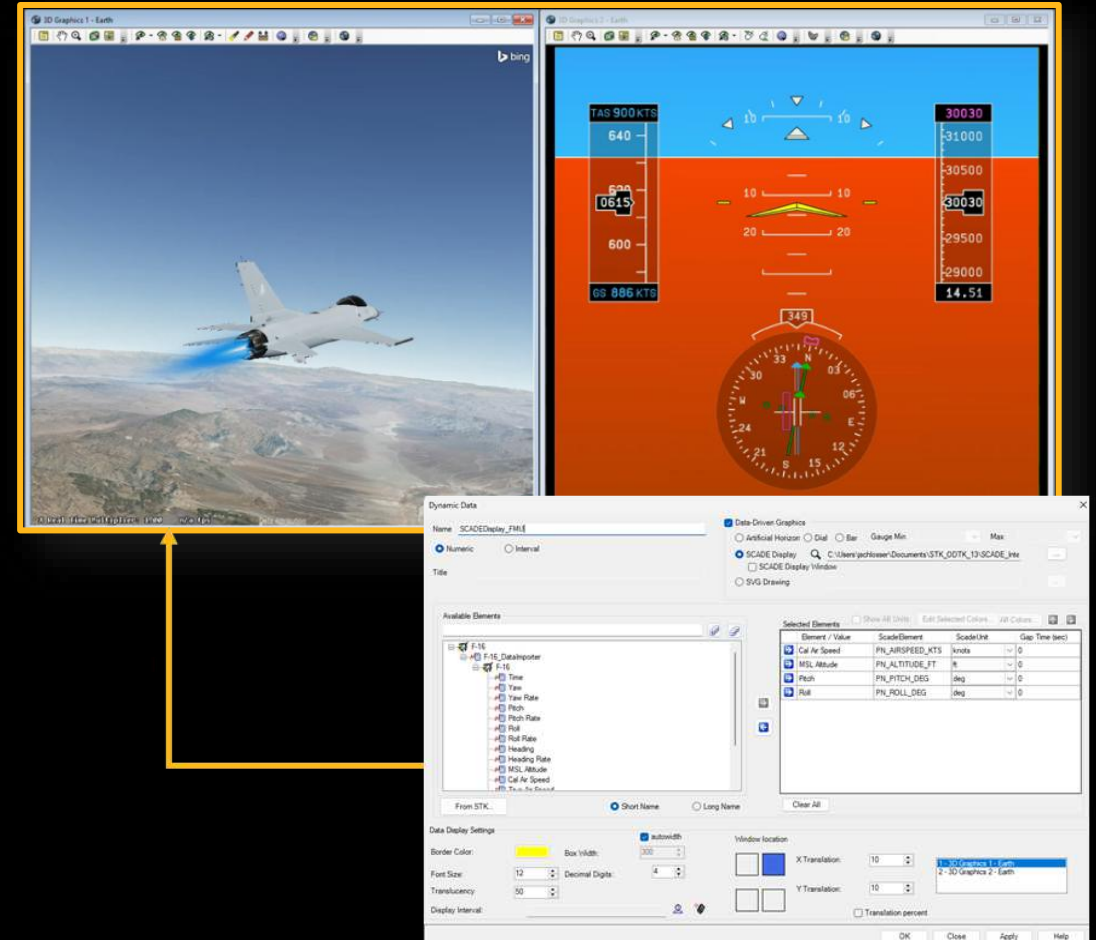


Drive Ansys SCADE Display with STK

Software Systems Engineering Software-in-the-Loop Simulation

Ansys STK users can now easily prototype, design, and integrate displays from the Ansys SCADE Display® model-based human-machine interface (HMI) design tool via exported FMUs. Engineers prototype and design in SCADE Display, export to FMU, then map STK variables to corresponding SCADE Display elements. This integration helps to reduce development time and live testing. It also supports the validation of performance in situations that are difficult or dangerous to perform with live testing.

- Export SCADE Display designs as FMUs
- Map STK variables to SCADE Display elements
- Data from STK scenarios drive the display with relevant mission data



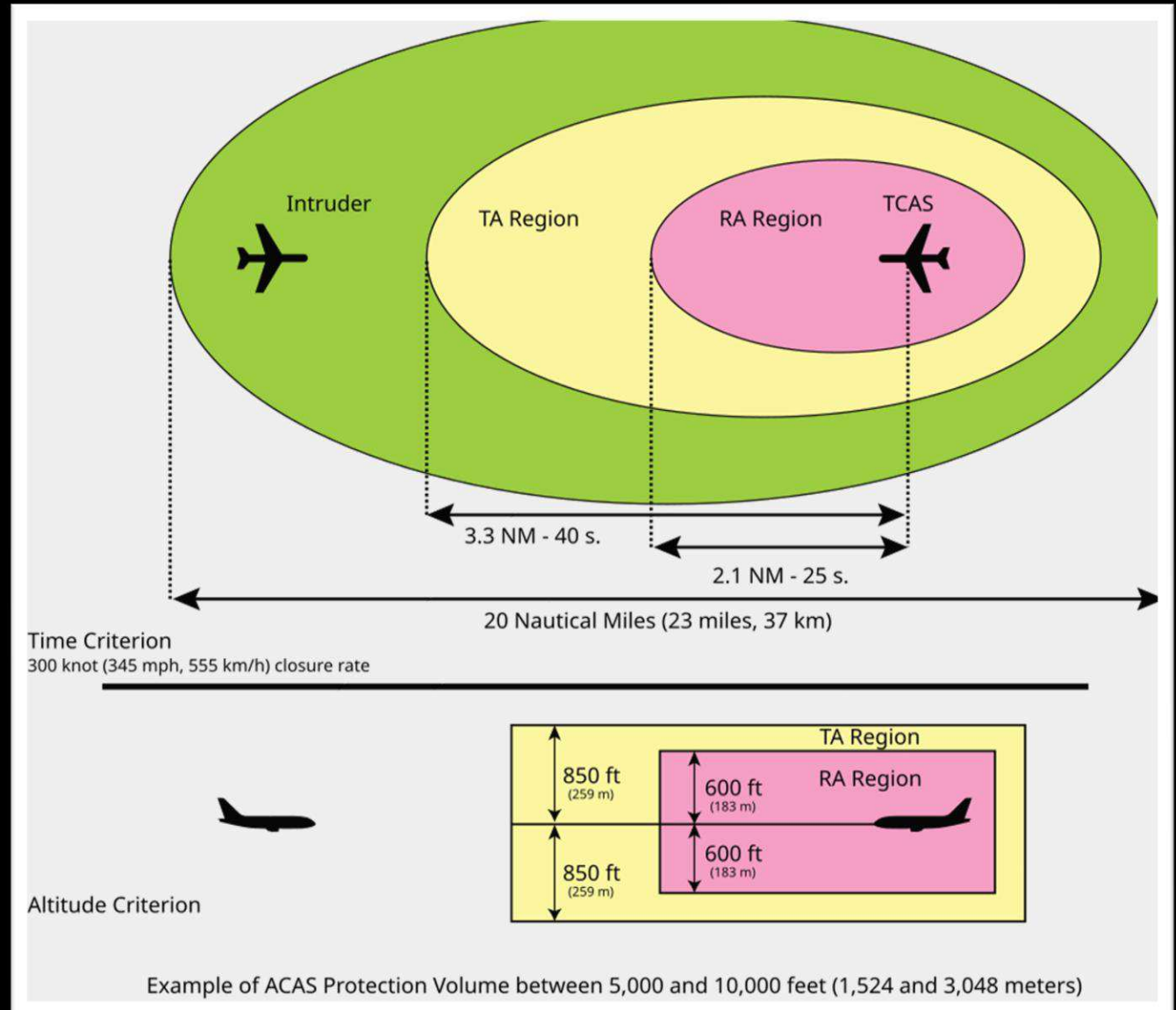
Demo Use Case: “TCAS” Traffic Collision Avoidance System

Problem: Avoid mid-air collisions

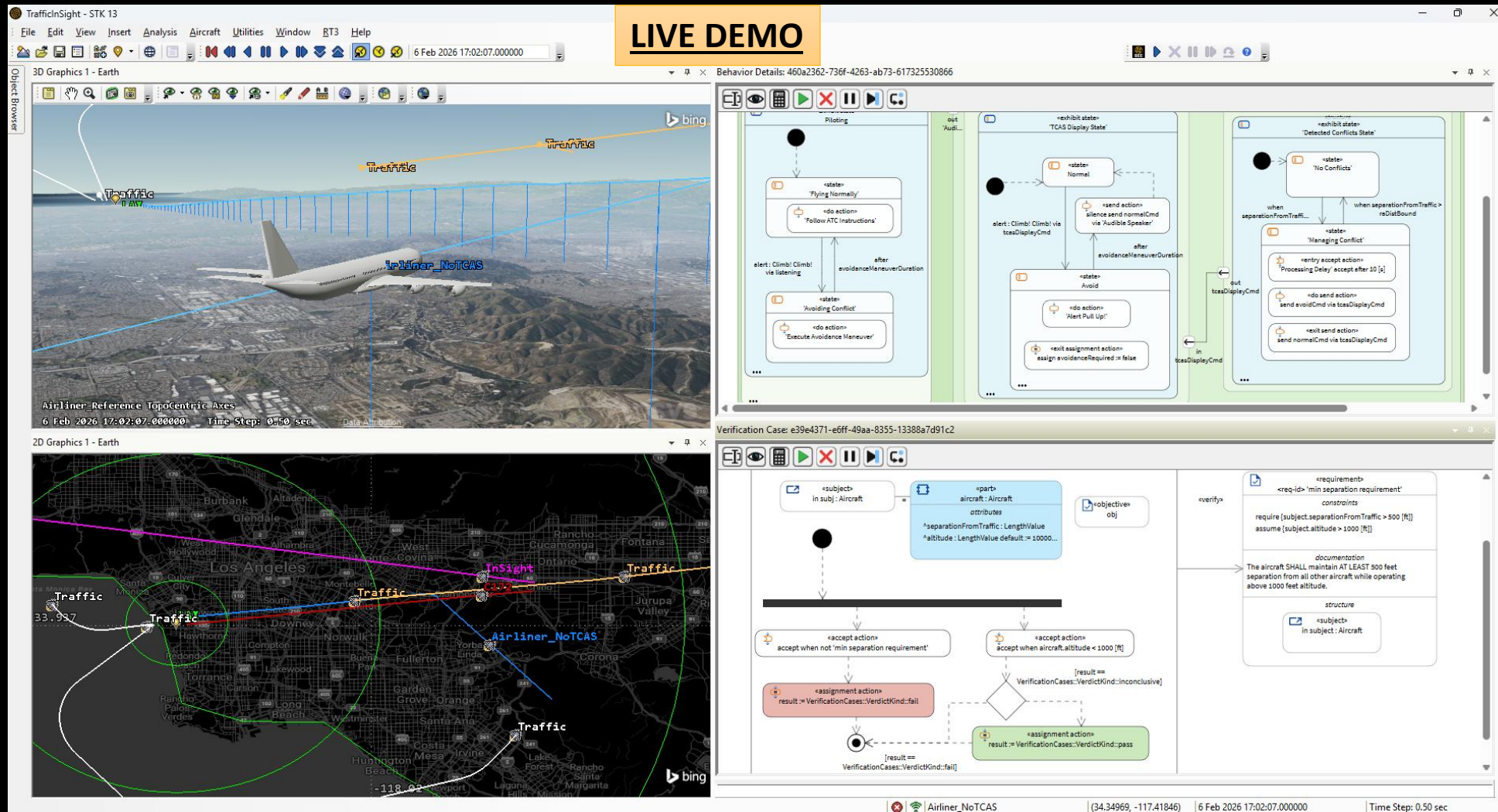
Proposal: Automatic Cockpit Alert System

Design: Use transponders to provide range alerts

... and now we need to verify it works in operation!



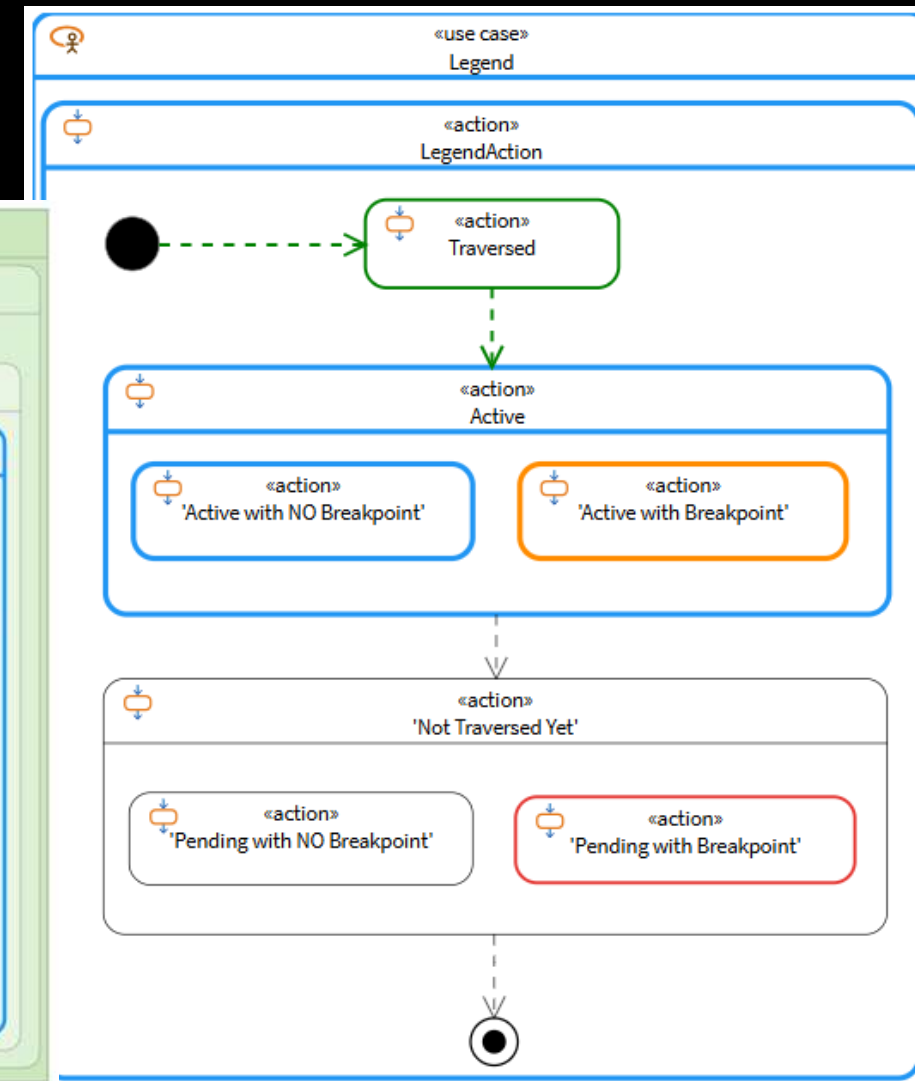
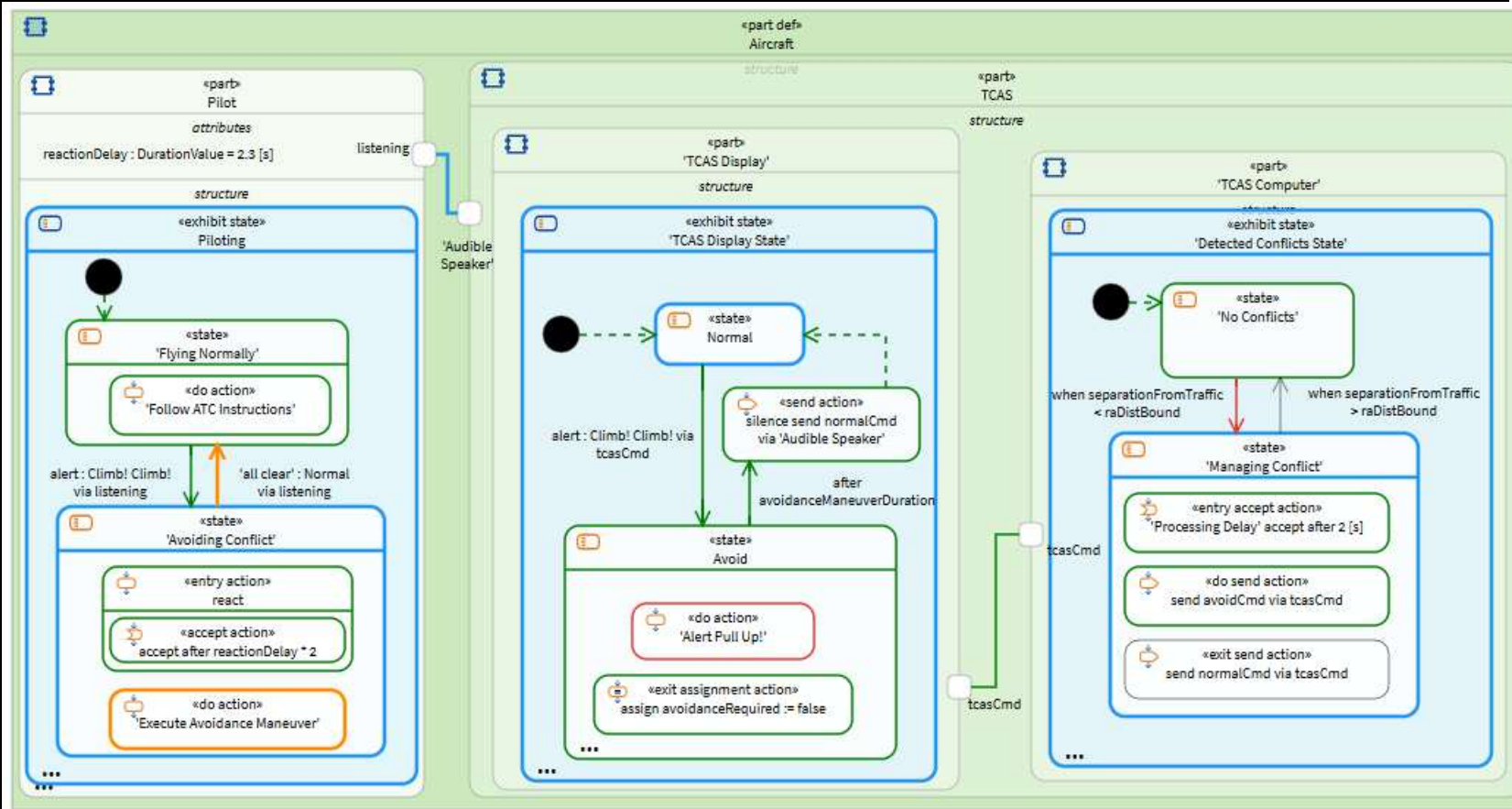
TCAS Operational Requirement Verification – Technology Demo



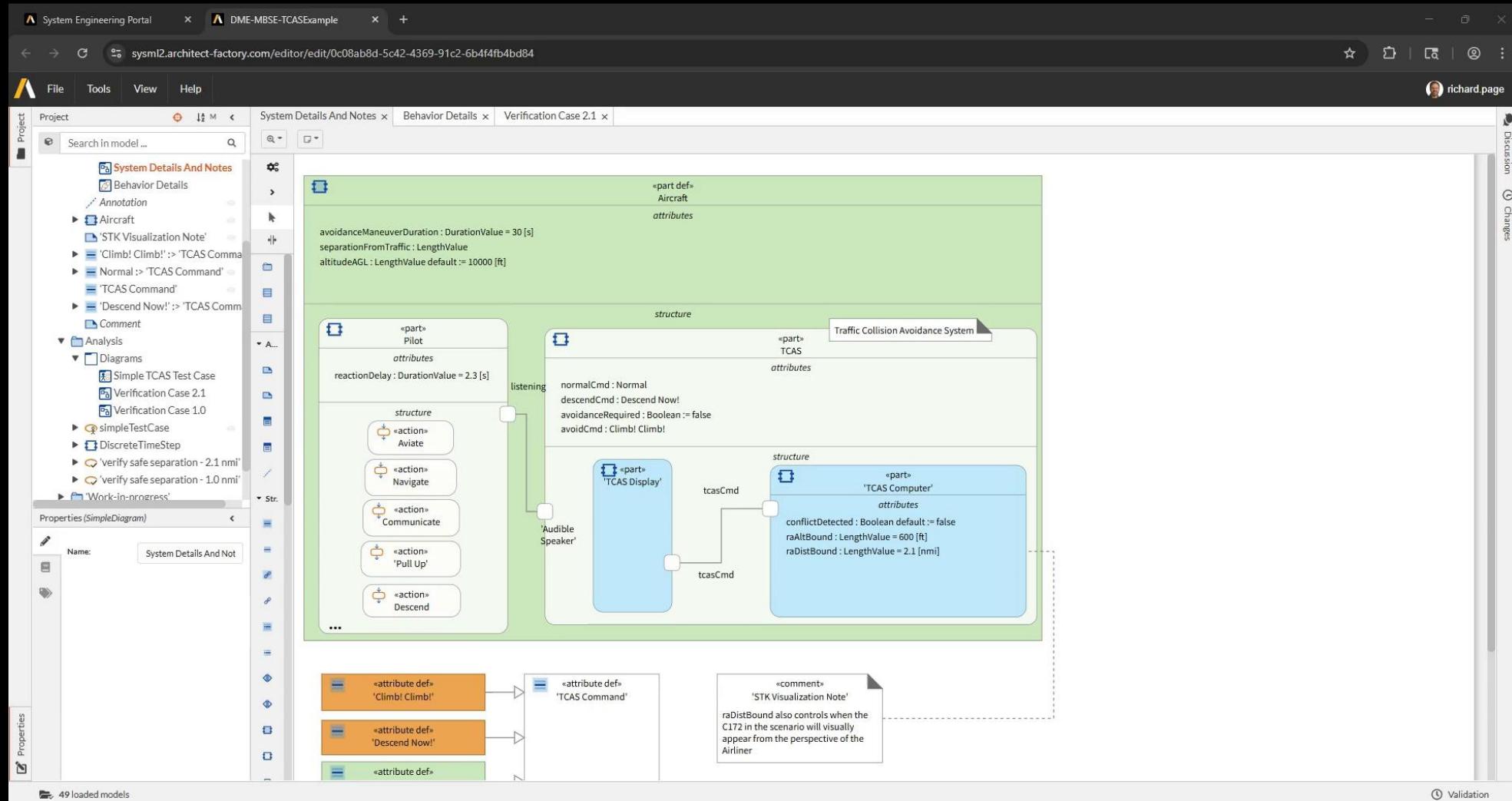
• For video, contact: rpage@agi.com

SysML View Execution Highlighting and Debugging

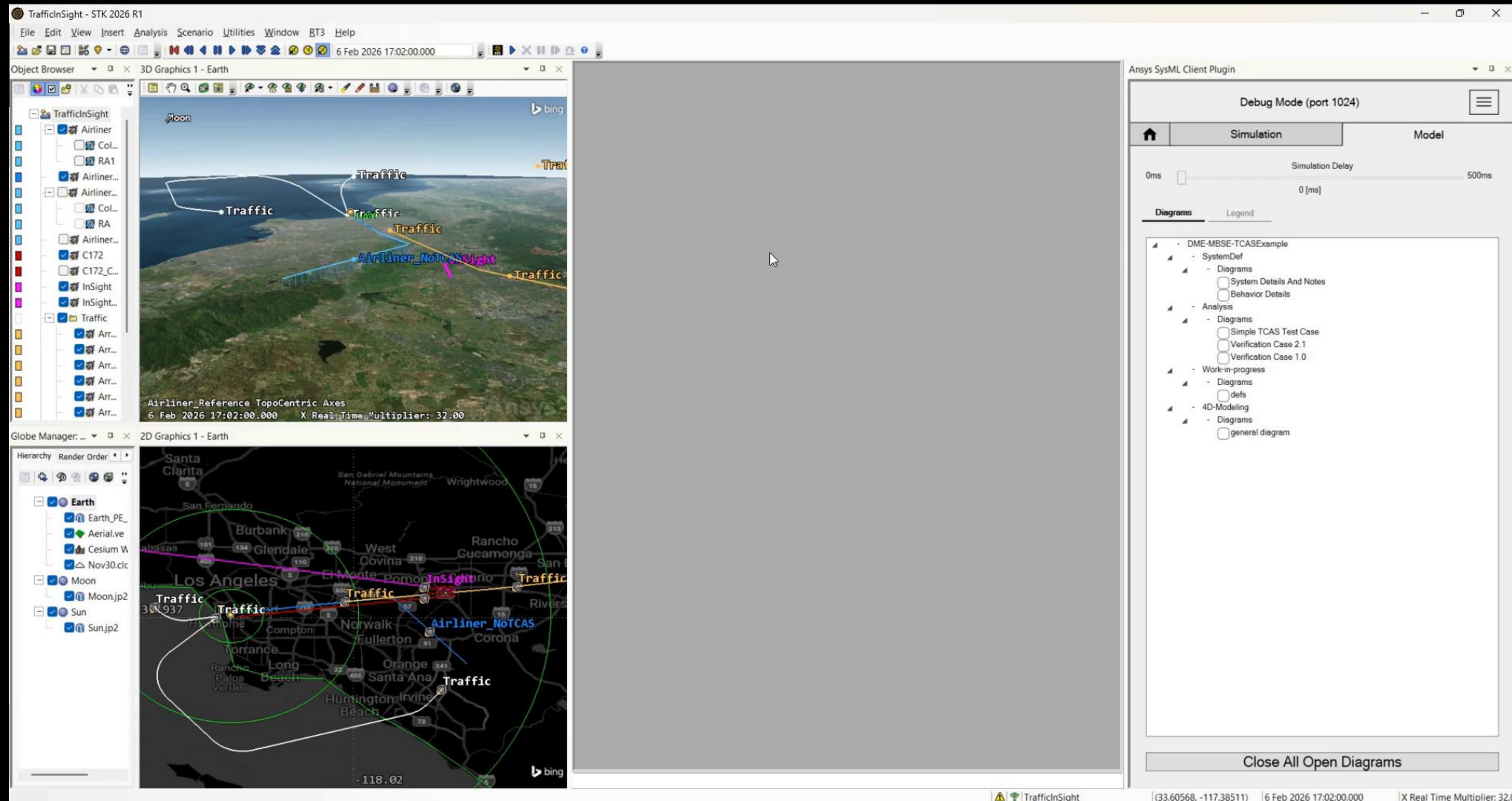
- Configurable coloring for highlighted elements



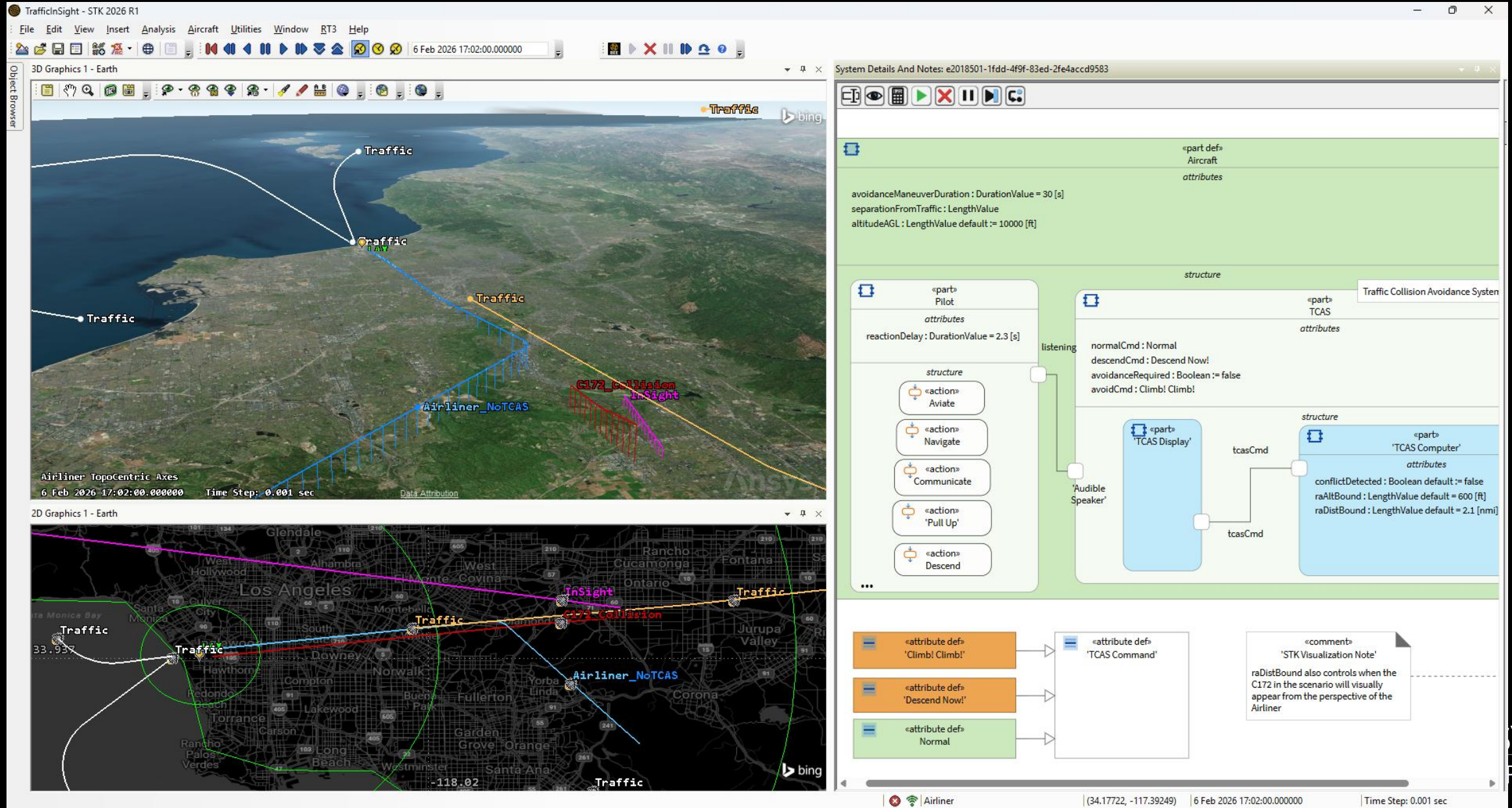
Modeling the Aircraft & TCAS for Execution



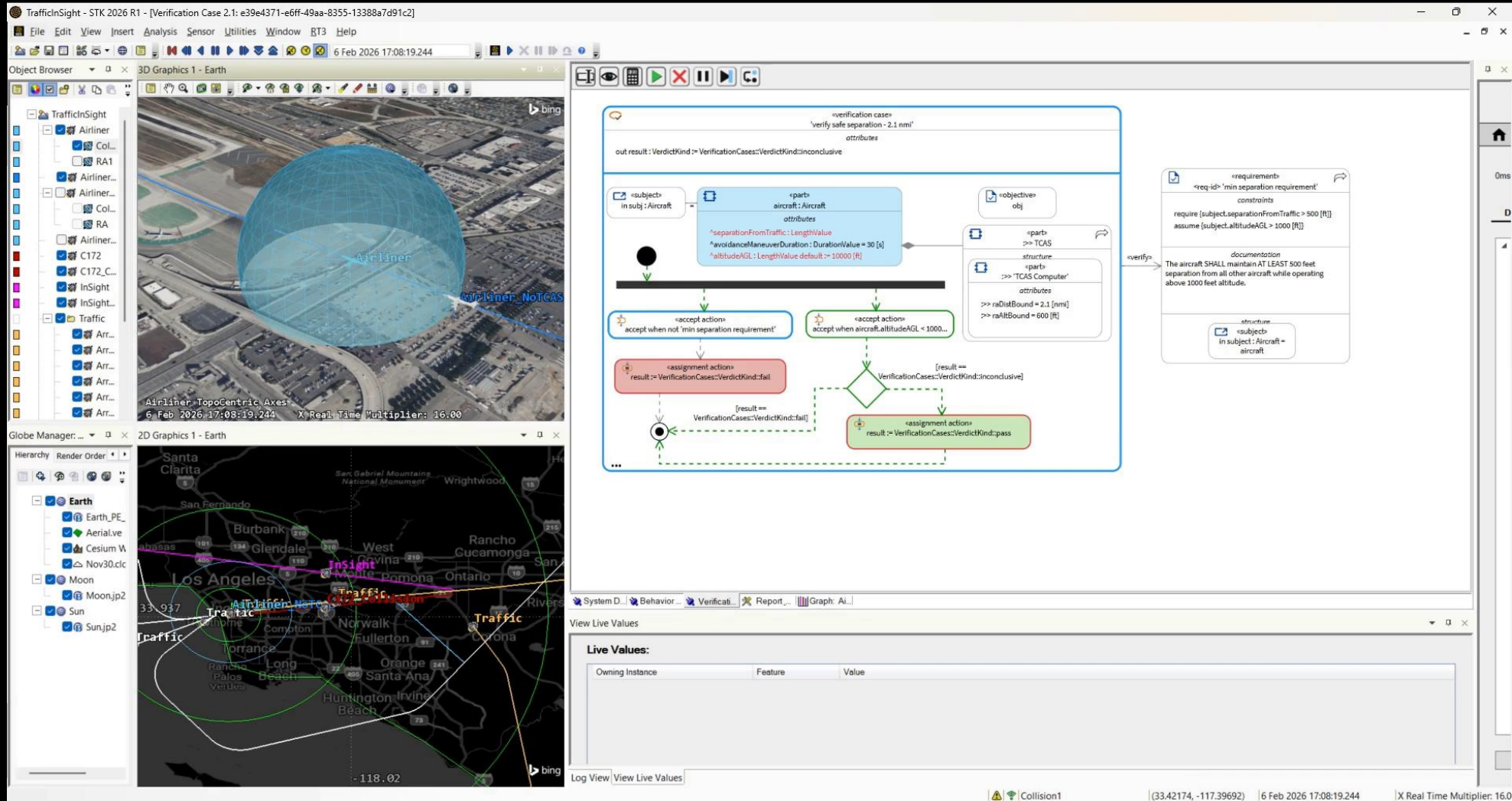
STK Design Reference Mission: Landing in LAX ... separation < 500 [ft]



STK Design Reference Mission: Landing in LAX ... separation < 500 [ft]



Verification Case Failure Condition: Using Change Events for V&V



Capturing Exact Moment of Failure with STK ... within ½ millisecond

TrafficInSight - STK 2026 R1

File Edit View Insert Analysis Sensor Utilities Window RT3 Help

3D Graphics 1 - Earth

Object Browser

Airliner TopoCentric Axes

6 Feb 2026 17:03:50.953174 Time Step: 0.001 sec

Airliner

Airliner Notices

6 Feb 2026 17:03:50.953174

“Execute Avoidance Maneuver” Modeled in STK Aviator

Configurable Systems Engineering “Dashboard”

Behavior Details: 460a2362-736f-4263-ab73-617325530866

«state» 'Flying Normally'

«do action» 'Follow ATC Instructions'

alert: Climb! Climb! via listening

«state» 'Avoiding Conflict'

«entry action» react

«accept action» accept after reactionDelay * 2

«do action» 'Execute Avoidance Maneuver'

«state» Normal

«send action» silence send normalCmd via 'Audible Speaker'

after avoidanceManeuverDuration

«state» Avoid

«do action» 'Alert Pull Up!'

«exit assignment action» assign avoidanceRequired := false

«state» 'No Conflicts'

when separationFromTraffic < raDistBound

«state» 'Managing Conflict'

«entry accept action» 'Processing Delay' accept a

«do send action» send avoidCmd via tcas

«exit send action» send normalCmd via tca

Verification Case 2.1: e39e4371-e6ff-49aa-8355-13388a7d91c2

«requirement»

<req-id> 'min separation requirement'

constraints

require {subject.separationFromTraffic >= 500 [ft]}

assume {subject.altitudeAGL > 1000 [ft]}

documentation

The aircraft SHALL maintain AT LEAST 500 feet separation from all other aircraft while operating above 1000 feet altitude.

structure

«subject»

in subject : Aircraft = aircraft

Live values during execution

Live Values:

Owning Instance	Feature	Value
'verify safe separation - 1.0 nmi'1'aircraft'1	separationFromTraffic	499.945219895452 [ft]

Requirement Violation

Diagram Visibility Settings: Behavior Details: 460a2362-736f-4263-ab73-61732553...

Instance Filter:

Created Instances

- ✓ 'verify safe separation - 1.0 nmi'1'aircraft'1/TCAS'1/TCAS Computer'1
- ✓ 'verify safe separation - 1.0 nmi'1'aircraft'1/TCAS'1/TCAS Display'1
- ✓ 'verify safe separation - 1.0 nmi'1'aircraft'1/TCAS'1/TCAS Pilot'1
- ✓ 'verify safe separation - 1.0 nmi'1'aircraft'1/TCAS'1/TCAS Computer'1/tcasCmd'
- ✓ 'verify safe separation - 1.0 nmi'1'aircraft'1/TCAS'1/TCAS Display'1/tcasCmd'1
- ✓ 'verify safe separation - 1.0 nmi'1'aircraft'1/TCAS'1/Audible Speaker'1
- ✓ 'verify safe separation - 1.0 nmi'1'aircraft'1/Pilot'1/listening'1

Filter which instances shown

Verification Case 1.0: 5c04a83a-8a34-473f-9292-d1c7b088ad78

«accept action» accept when not 'min separation requirement'

«accept action» accept when aircraft.altitudeAGL <...

«assignment action» result := VerificationCases::VerdictKind::fail

Breakpoint on Failure

Simulation “Trace” Log

Log View

Level	Time	Message
Information	2026-03-27 14:09:05	<(2026-02-06T17:03:40.893421571Z, DelegatedActionStepped(ActionUsage!DME-MBSE-TCASExample::SystemDef::Aircraft::Pilot::Piloting::Avoiding Conflict::Execute Avoidance Maneuver!519ada96-2262-4858-904a-d7fc54cd7782))>
Information	2026-03-27 14:09:05	<(2026-02-06T17:03:40.893421571Z, TransitionStarted(TransitionUsage!DME-MBSE-TCASExample::SystemDef::Aircraft::Pilot::Piloting::!80a687df-961d-4969-b54f-78c258dde72e))>
Information	2026-03-27 14:09:07	<(2026-02-06T17:03:40.893421571Z, ActionStarted(AcceptActionUsage!H0720d7e-ffd8-40d9-be33-c407c23d07c6))>
Information	2026-03-27 14:09:07	<(2026-02-06T17:03:50.953173870Z, ActionDone(AcceptActionUsage!81db9d4c-0d68-4337-b9c1-8020fb9dd5e))>
Information	2026-03-27 14:09:07	<(2026-02-06T17:03:50.953173870Z, ActionStarted(AssignmentActionUsage!d01e67a4-f7f8-49c4-b54f-5d1321e833a1))>

Behind the Scenes: Tool Connection API

Behavior Execution API for Custom Behaviors

- Java API for custom code

- Actions
- ScalarValues
- Custom Event Detection
- <https://help.agi.com/bee>



- “pySTK” available!

- <https://docs.pyansys.com/>
- <https://stk.docs.pyansys.com/>

- SAM Supports

- Standard SysML v2 API
- SAM API for scripting



- More Python coming soon!

```
@Override
public void registerDelegate(CustomCodeRegistry customCodeRegistry) {
    try {
        String aircraftId = "DME-MBSE-TCASExample::SystemDef::Aircraft";

        customCodeRegistry.registerInstanceDataFor(
            aircraftId,
            TcasAircraft.class,
            InstanceContext instanceContext -> new TcasAircraft(instanceContext, mStkRoot, aircraftId)
        );

        customCodeRegistry.delegateExecutionOfAction(
            s: aircraftId + "::Pilot::Piloting::Avoiding Conflict::Execute Avoidance Maneuver",
            ActionContext actionContext -> {
                TcasAircraft ac = actionContext.getDataService().getDelegationData(TcasAircraft.class);
                return ac.executeAvoidance(actionContext);
            }
        );

        customCodeRegistry.delegateScalarValueForFeature(
            s: aircraftId + "::altitude",
            new AltitudeAccessor()
        );

        customCodeRegistry.delegateScalarValueForFeature(
            s: aircraftId + "::separationFromTraffic",
            new MinSeparationAccessor()
        );
    }
}
```

Simple Registration
to “delegate” behavior
from SysML to code

Code once for reusable SysML modules!

SysML 2 Execution Implementation Reflections

- **Execute SysML models to enable:**
 - Connection to non-SysML Authoritative Digital Artifacts (e.g. STK Design Ref. Mission)
 - Operational Requirements Verification & Validation
 - Debugging and Iterating on Complex Designs for Systems-of-Systems
- **Needs from SysML 2.x ++ (Leading in the OMG / SMC / Execution Working Group)**
 - Better Time Library (time.sysml) support for time operations (e.g. “DurationBetween”)
 - Need to specify an epoch (TimeInstantValue) to start clocks for execution
(i.e. No easy and obvious way for modelers to specify this besides “localClock.currentTime := ?”)
 - Better expressive guidelines for “flows” (of packages? Of water? Of RF Comms?)
 - Decibel logarithmic unit dimension support (for RF Comms Link Budgets)

Plans for Systems Modeling Community: Execution Working Group

- Publish “Initial Guidance” by end of 2026

- “**Execution Rules**” – Heuristics for tools to execute undefined models in a useful way
- “**Minimal Time**” – Actions can take any time ... but let’s choose the shortest time
- “**Initiating Actions from Start**” – Actions may not execute by default ... but let’s say “at least once”
- “**Perform Actions**” – Actions on parts don’t run ... unless *performed*.
- (More rules under consideration)



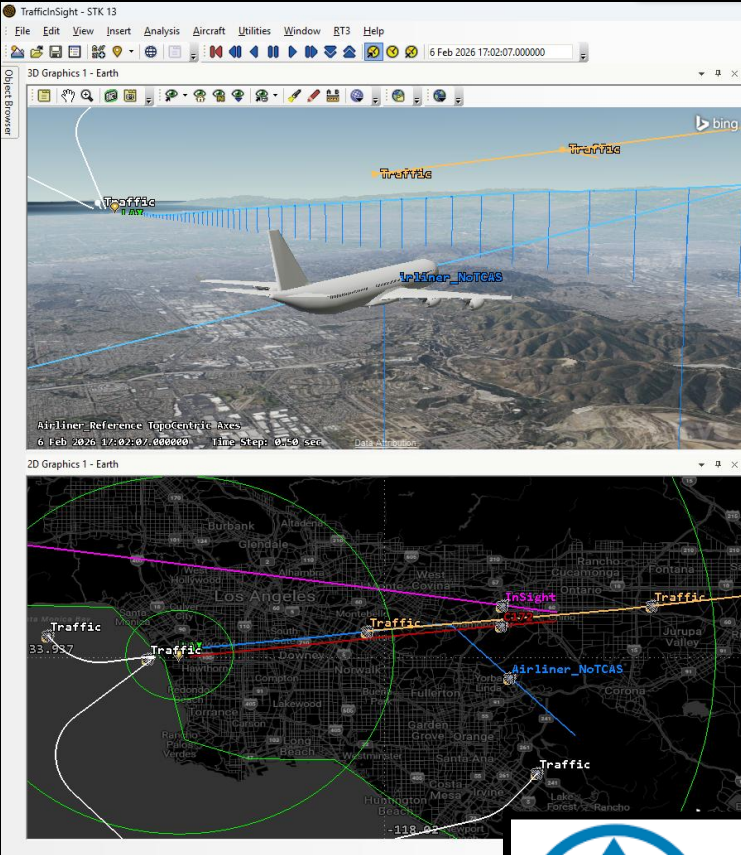
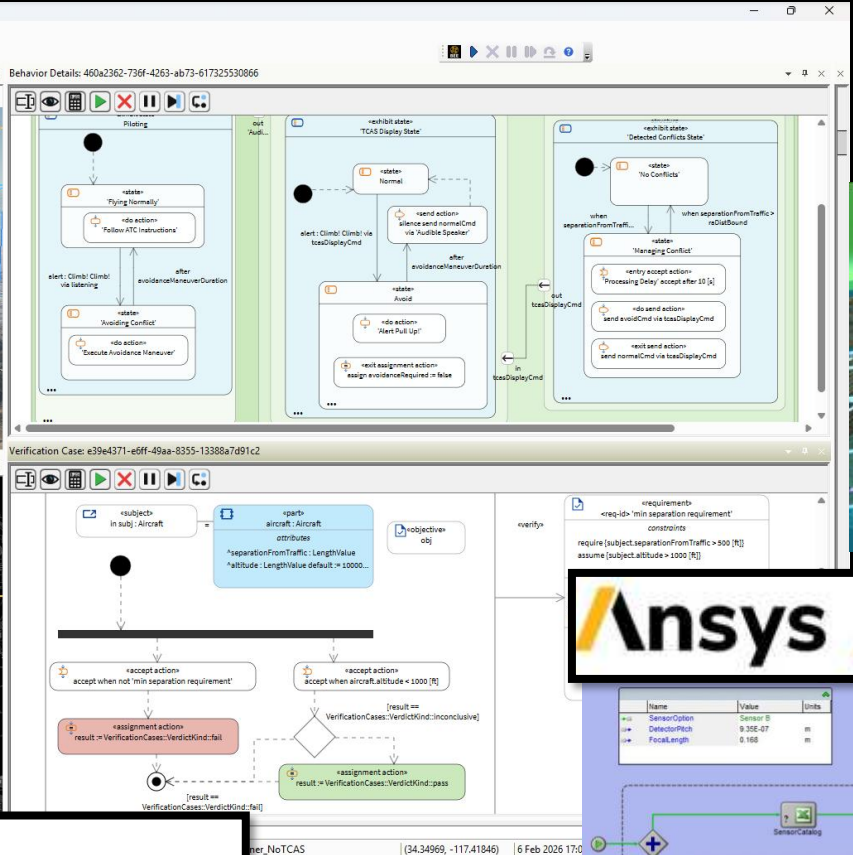
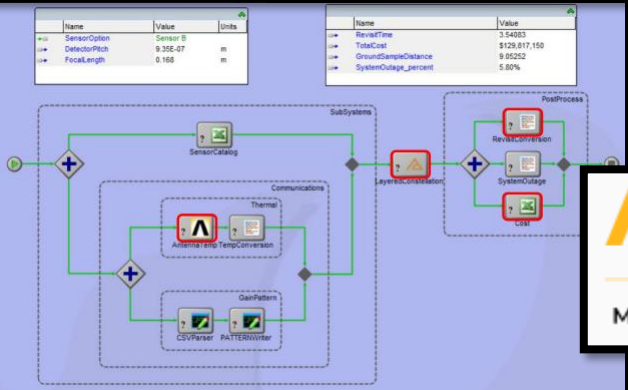
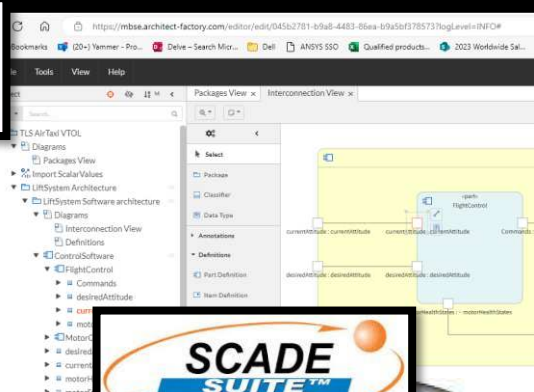
- Initial Scope of “Execution Cases”

- Basic actions
- Basic states
- Send/Accept and "Dispatch" semantics
 - Including sending messages across interfaces
- Perform/Exhibit
 - “Calling” behaviors
 - Structure *performing* functional behavior
- Flows
 - Discrete item flows

Will be available on public GitHub:

<https://github.com/Systems-Modeling/SysML-v2-Release>

Ansys provides many ways to connect to and *Execute* SysML v2.x



Thank you!



**For questions or software requests, contact:
richard.page@synopsys.com**