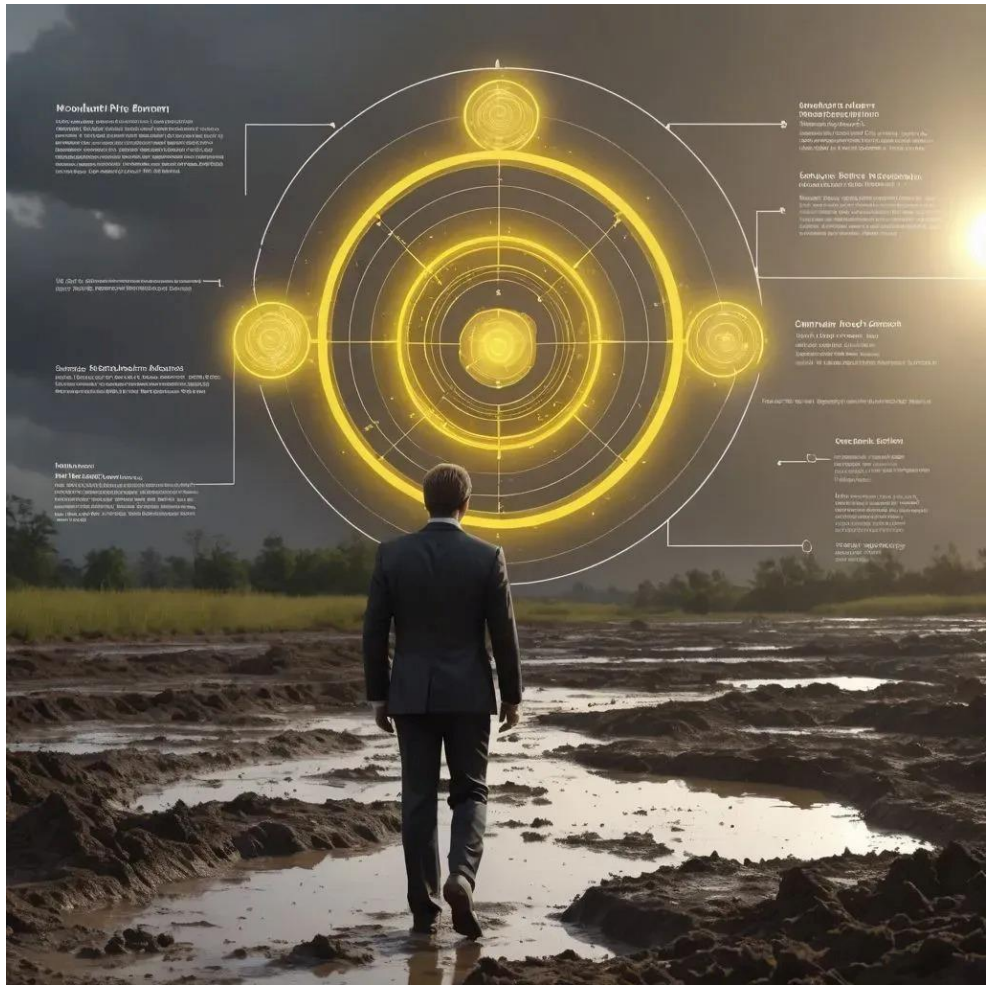
Advanced MBSE with SysMLv2

Mike Crist - Siemens

Presenters Bio

Mike Crist is a recent add to the Siemens Digital Industries Software Division after spending 16 years at Raytheon Technologies. His focus is on Model Based System Engineering, enabling the Aerospace, Defense, Federal and Marine customers to adopt these technologies. Mike is working on his PhD in Systems Engineering from Colorado State University and his dissertation material addresses ISO 10303-243 / STEP AP243.

The slog of systems engineering



As a model-based systems engineer, I can dream up beautiful, intricate solutions to problems…

…that I have to describe with SysMLv1 and other engineering languages…

…sometime with an architectural framework…

…sometimes with automation…

…but it take so much time and effort to create every nit-picky detail and link everything together in the tools today…

…and even if I did describe my beautiful system, no one else on my team could efficiently use my work!

The slog of systems engineering

So I stick with making detailed block diagrams with proper element types to represent my solution…

…but I also need just a few custom stereotypes…

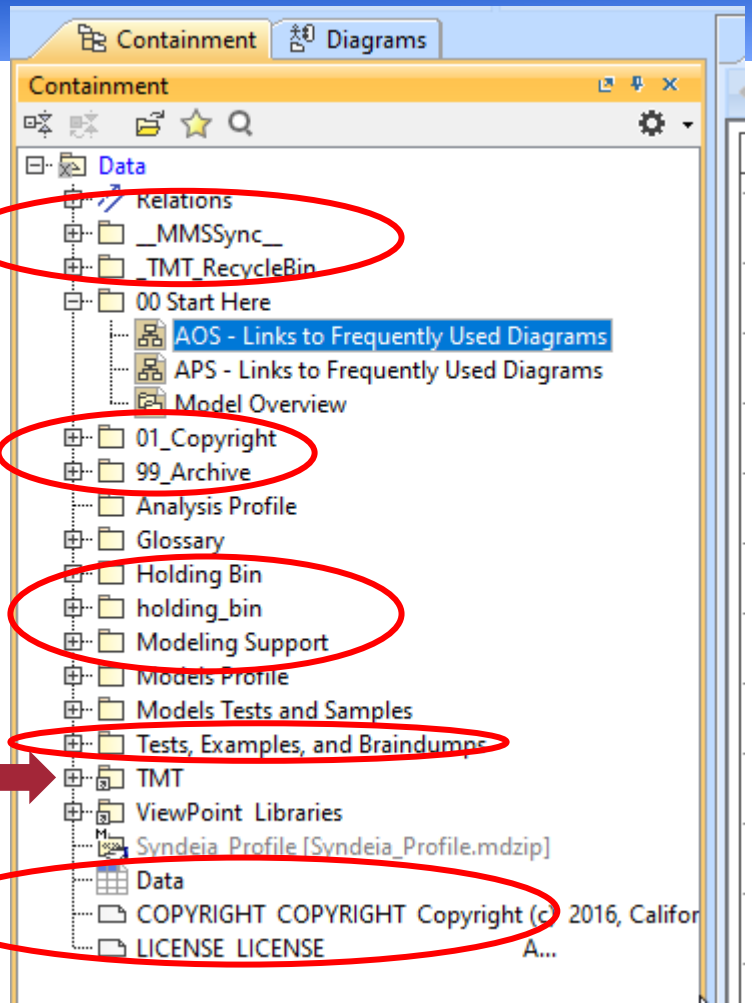
…and just a few custom tool automations…

…and maybe just a few custom tables and matrices…

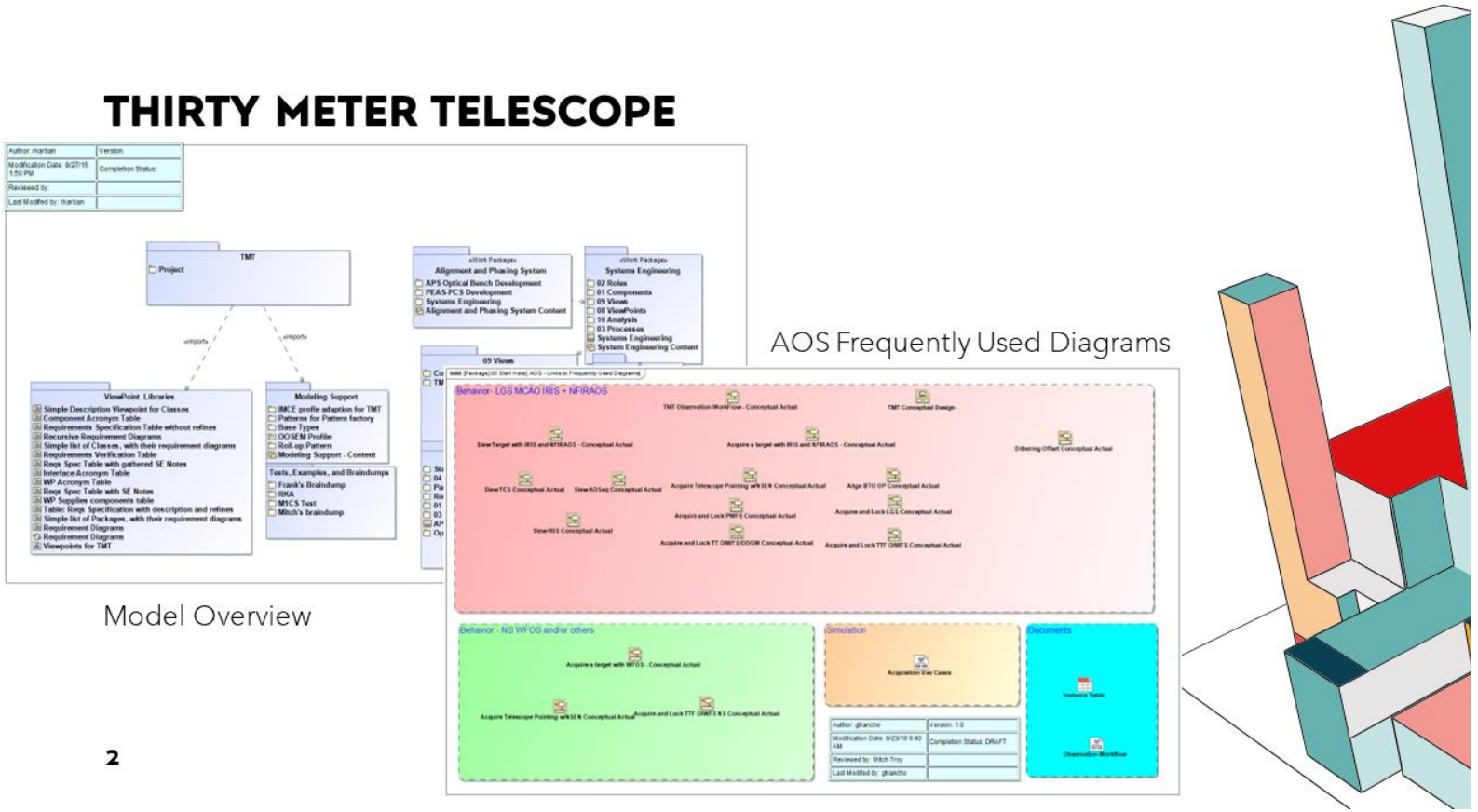
…and I should probably store some notes about what I did…

…and I do all of this by *myself*…

The design is in here



The slog of systems engineering



So I can export them to pictures I store in Powerpoint to communicate to the rest of the team.

Bedraggled and weary, the lead systems engineer wanders into a large cavern.

There in the middle of the cave, sitting atop three bloodied bodies of his IPT leads is the Unfunded Mandate Dragon!

Defiantly he cries “My SysML model IS a model based digital twin! What more do you want from me?!?!”

The dragon just laughs, and in a low rumble, says ”*Verify the implementation meets your requirements.*

The lead systems engineer panics as he suddenly realizes he didn’t import this weeks Excel spreadsheet of performance parameters into the model!

He flips through his Powerpoint deck of spells looking for some way to defend himself

And then the nightmares come

What if I have to change the model?!?!!

What if the implementation doesn’t match?

What if the customer doesn’t like it?

What if the design is incomplete?

What if it can’t be implemented?

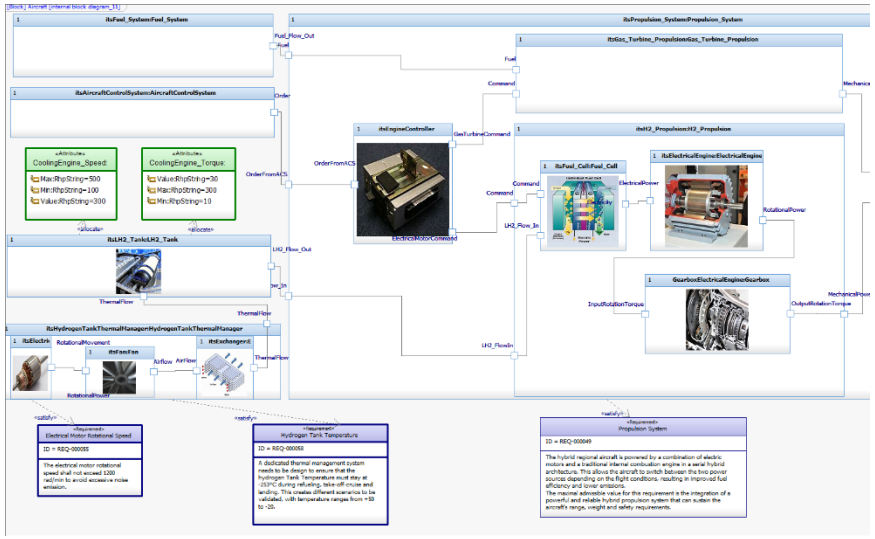
We don’t want to do this anymore

Partnership with IBM

- We know SysMLv1 is nearing a dead end, but need to provide continuity to what's next

SIEMENS + IBM

- **Best-in-class SysML v1 integrated with PLM**
- **Delivering system to software continuity**
- **Enabling early concept design verification**
- **<u>Developing best-in-class SysML v2 solution</u>**



Engineering teams succeed when they collaborate across multiple domains and tools based on connected data

Example

Software Architect

- Creates SW architecture
- Specifies behavior on external interface level

Software Developer

- Adds internal SW behavior
- Implements code

SW Release to ECU/LRU integration

Report on feature progress

Systems Engineer

- Notified to design a new feature
- Designs & simulates multi-domain architecture

E/E Architect

- Designs E/E platforms & topologies
- Allocates functions

Network Engineer

- Configures the network
- Runs automated gateway configuration

ECU / LRU Integrator

- Configures ECU / LRU middleware
- Integrates applications

ECU/LRU Configuration release to implementation

Report on E/E topology progress

Mechanical Engineer

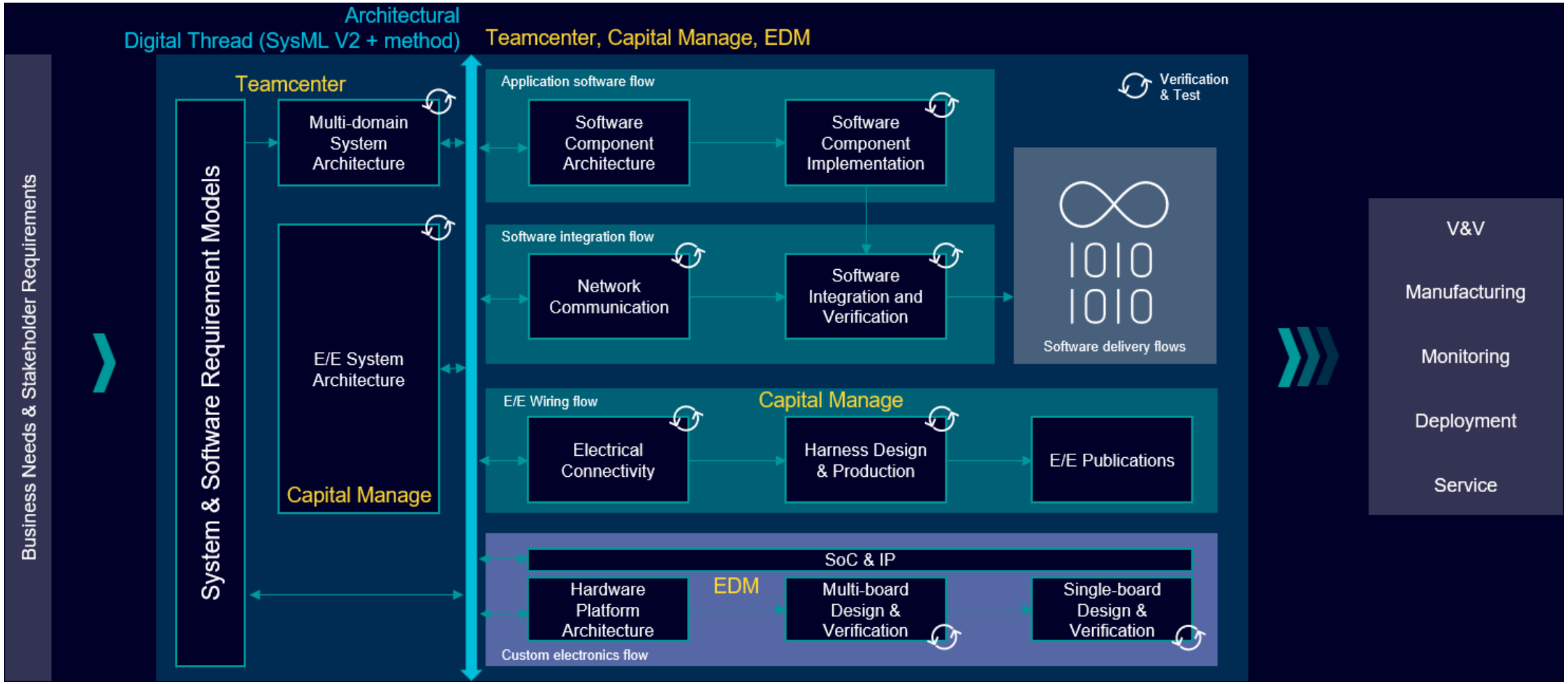
- Develop BOM utilizing trade studies leveraging functional and non-functional system constraints

CAE Specialist

- Leverages system architecture to drive multi physics simulation

Feedback & process loops

Digital Thread Accessibility Libraries – Putting the DigiTAL in Digital Thread



SysML v2 – A new representation of a model

- Consistent usage / definition pattern across all ele t types

  Names are consistent too!
- Good features and practices a lan

  Function chains

  Imports / usage f

  Separation of mode

  Multiple views based o

  Calculation and equatio the mo

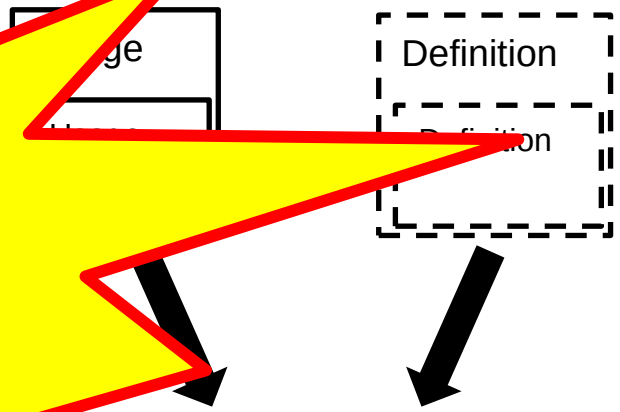
  Formal descriptio verification

  Textual and graphica

  Domain specific
- REST API is defined as part d dard

  Create / Read / Update / Delet ery the model using commands defined in the st dard

No vendor lockin!



	SysML v2
	Part / Part Definition
Value Type	ibute / Attribute Definition
ce Block	Port / Port Definition
Action / Activity	Action / Action Definition
State / State Machine	State / State Definition
Constraint Property / Constraint Block	Constraint / Constraint Definition
Requirement	Requirement / Requirement Definition
Connector / Association Block	Connection / Connection Definition
View	View / View Definition

Imandra brings Automated Reasoning to any SysML v2 Environment

MBSE Tool

SysML v2 Model

- Robust intuitive tools
- Define, track, and visualize all aspects of systems
- Engineering analysis for design evaluation and requirements tracking.

Transpile (Convert)

Return results for visualization and integration

Imandra

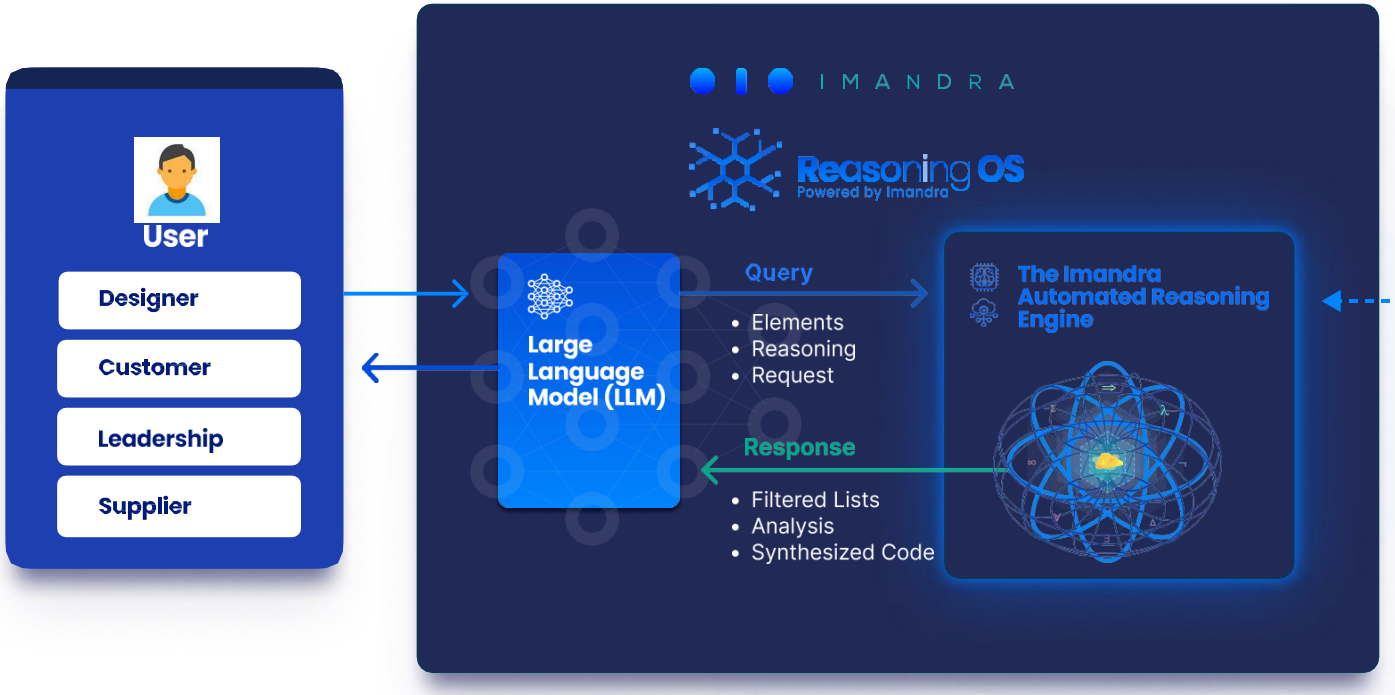
IMANDRA

Imandra Modeling Language

- Model checking of State-machines
- Formal proofs of SysML v2 Constraints and Requirements
- Formal impact analysis of changes
- Trade space analysis
- Hallucination-free ChatBot and Formal Reasoning using a Natural Language Interface

Imandra integration with LLMs

Imandra is able to connect a natural language interface to your SysML v2 model to allow a broader set of stakeholders to understand the design, ask questions, and perform critical tasks without design team intervention or inclusion.

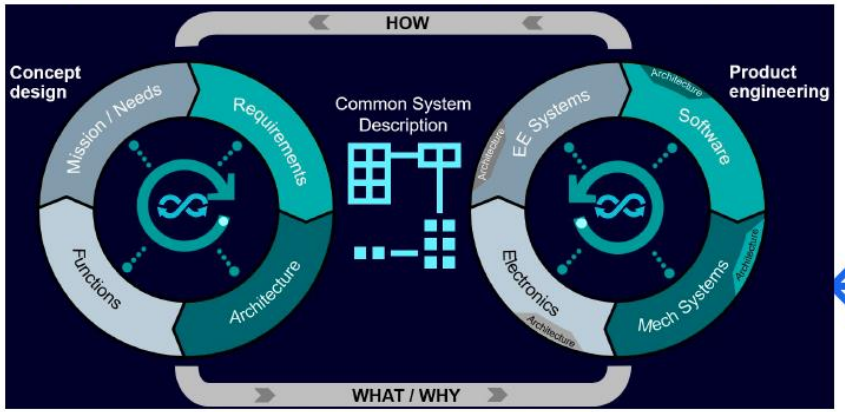


Conversion

Automatically transpile a SysML v2 Model and Connect to a Hallucination-Free Natural Language Interface (ChatBot)

Siemens TeamCenter Integration with the Imandra

Siemens TeamCenter with SysML v2 for MBSE
(Visual Lifted form Siemens SysML v2 White Paper)



Automatic Conversion from SysMLv2 to Imandra Using the SysML v2 API and our Transpiler

Imandra Automated Reasoning (based on symbolic AI)

- Formal verification
- Complex system proofs
- Procedure analysis
- Model refinement and integration from co-simulation, physical test, and telemetry
- System exploration and discovery
- Trade-space instance synthesis and optimization
- Natural language integration

Conceptual Example of LLM Use Cases

Show the parts in the EPS that impact heating?

Sure, here are the parts...

What are the events that would cause a thruster to fire?

Here is a list of states and attributes that lead to a...

Show me all verification cases tied to the trans-lunar phase of Artemis I,II,& III

Here are the verification cases for...

Add a requirement that all tanks must be pressure tested every 30 days.

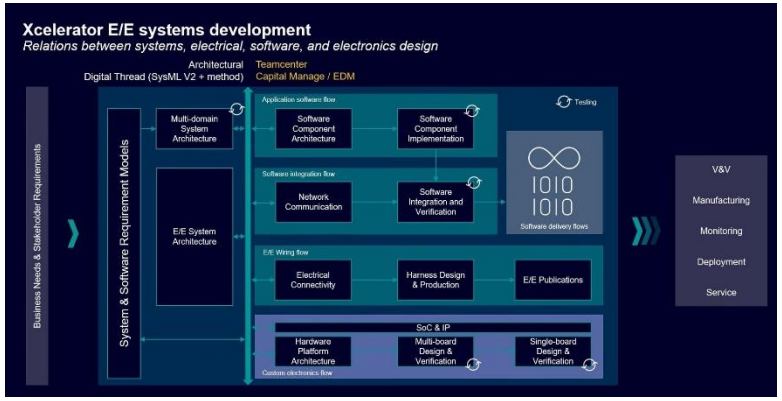
Here is the requirement in SysMLv2 in textual notation...

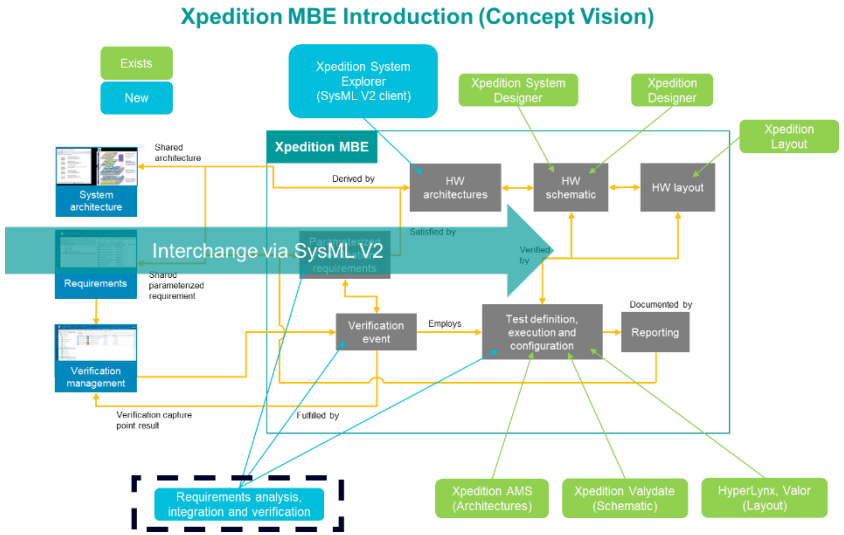
LLM Integration

- Ask questions about the SysMLv2 model
- Synthesize SysMLv2 (requirements, structure, behavior)
- Hallucination-free results with audit trail
- Enable more stakeholder to collaborate

Xpedition MBE - Enabling the Comprehensive Digital Twin for Electronic Systems

Global Product Data Interoperability Summit | **2024**





Requirements traceability and verification

- Breakdown of requirements into PCB test cases
- Association of test cases to execution
- Automation of execution
- Visibility to verification status through verification capture point

Continuous Architectural Breakdown

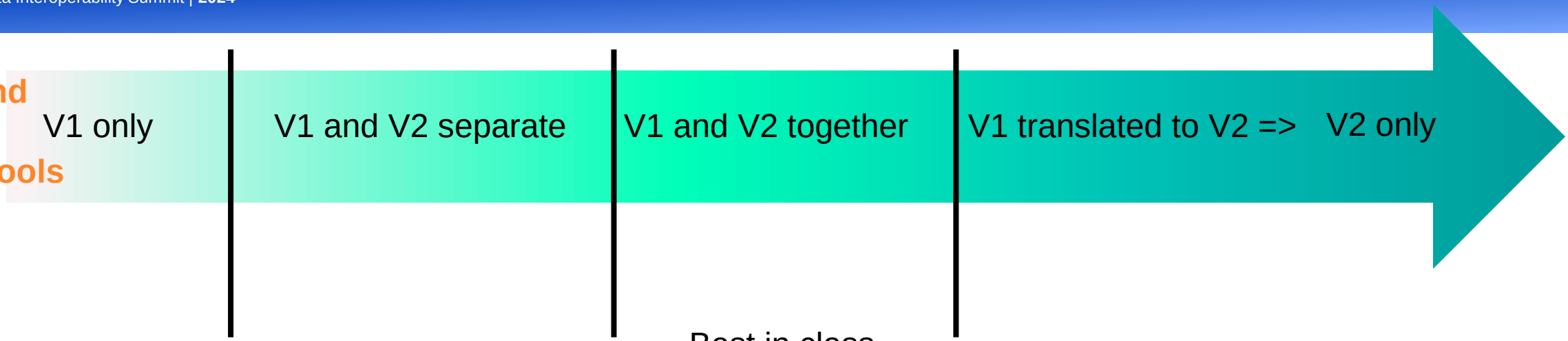
- Decomposition from system level to domain implementation
- Annotation of requirements onto architecture
- System and sub system level simulation

Key Multi-release investments

- Requirements breakdown and verification
- SysML V2 adapter
- SysML V2 Architecture integration into Xpedition design flow
- Industry standard Interfaces

SysML V1 to SysML V2 Continuity Strategy

Siemens and Third Party Modeling Tools



Best in class V1 integration

Cameo V1 to Rhapsody V1 translation

Support UAF, NAF, OA modeling

Common SE tool information model

OEM V2 Authoring

V2 import and export in Multiple domains

Best in class V1 and V2 integration

V2 Authoring Including V1 artifacts

OEM V1 and V2 Coexistance in models

Best in class V2 integration

OEM V2 Authoring

Server side translation

Support UAF V2

Targeted Release Schedule - System Modeling

Rhapsody
Cameo
SMW
SysML V2
Capital

Fall 2023 - Teamcenter 2312	**Spring 2024** - Teamcenter 2406	**Fall 2024** - Teamcenter 2412	**Spring 2025** - Teamcenter 2506
2023	**2024**		**2025**
Many tool / Multi-tool Common Information Model			
Systems Modeler with IBM Rhapsody 2402	**Systems Modeler with IBM Rhapsody 2406**	**Systems Modeler with IBM Rhapsody 2412**	**Systems Modeler with IBM Rhapsody 2506**
Rhapsody Integration 2312	**Rhapsody Integration 2406**	**Rhapsody Integration 2412**	**Rhapsody Integration 2506**
Cameo Integration 2312	**Cameo Integration 2406**	**Cameo Integration 2412**	**Cameo Integration 2506**
System Modeling Workbench 2312 Capella 6.1	**System Modeling Workbench 2406** Capella 7.0	**System Modeling Workbench 2412** Capella 7.0	**System Modeling Workbench 2506** Capella 7.1
Architect Explorer 2312 Capella 6.1 with MBCSE	**Architect Explorer 2406** Capella / SMW with MBCSE	**Architect Explorer 2412** Capella / SMW with MBCSE	**Architect Explorer 2506** ?? with MBCSE
Simcenter Studio V2 integration Studio 2210 and later, 2310 released	**Simcenter Studio** Studio 2404	**Simcenter Studio** Studio 2410	**Simcenter Studio** Studio 2504
	SysML V2 Modeler GA1 Joint development with IBM	**Teamcenter SysML V2 Integration 2412***	**Teamcenter SysML V2 Integration 2506**
	Capital Software Developer with IBM Rhapsody 2402	**System Lifecycle Modeler 2412*** Joint development with IBM	**System Lifecycle Modeler 2506** Joint development with IBM
		Capital Software Developer with IBM Rhapsody 2408	**Capital Software Developer with IBM Rhapsody 2502**

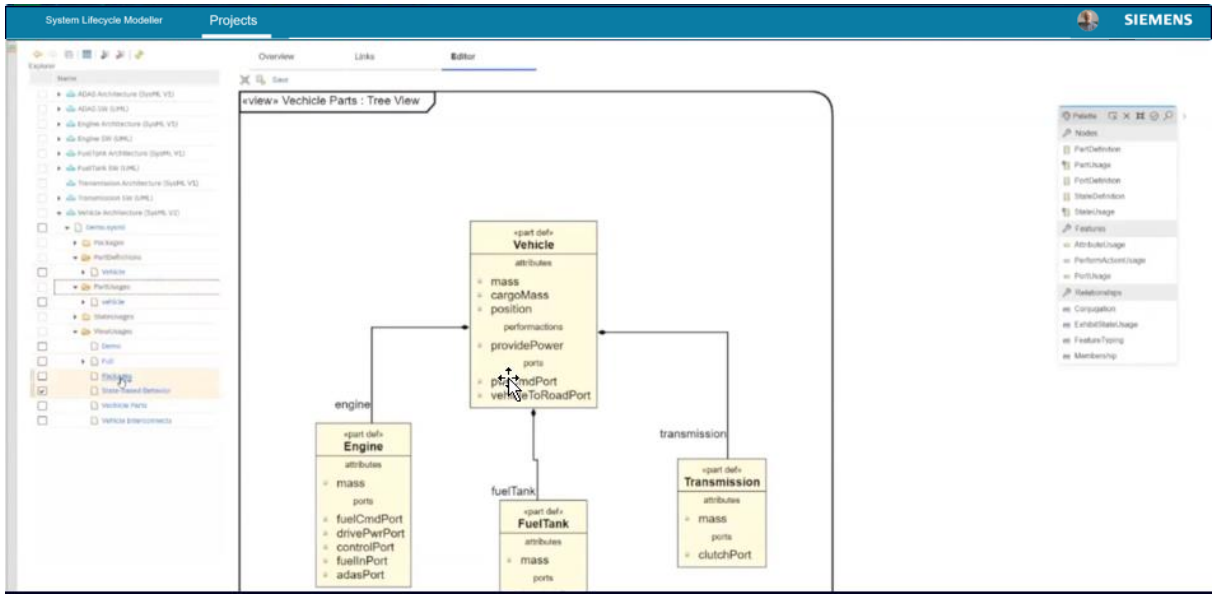
GLOBAL PRODUCT DATA INTEROPERABILITY SUMMIT 2024

* Stretch Goal

System Lifecycle Modeler

* Stretch Goal

2412*



Standalone MBSE web authoring supporting next generation SysML v2 standard 2412

- Capture product architecture with standard SysML V2 language
- Speed up your design by combining textual and graphical authoring with advanced visualization
- Collaborate on the system architecture across the enterprise with System Web Authoring

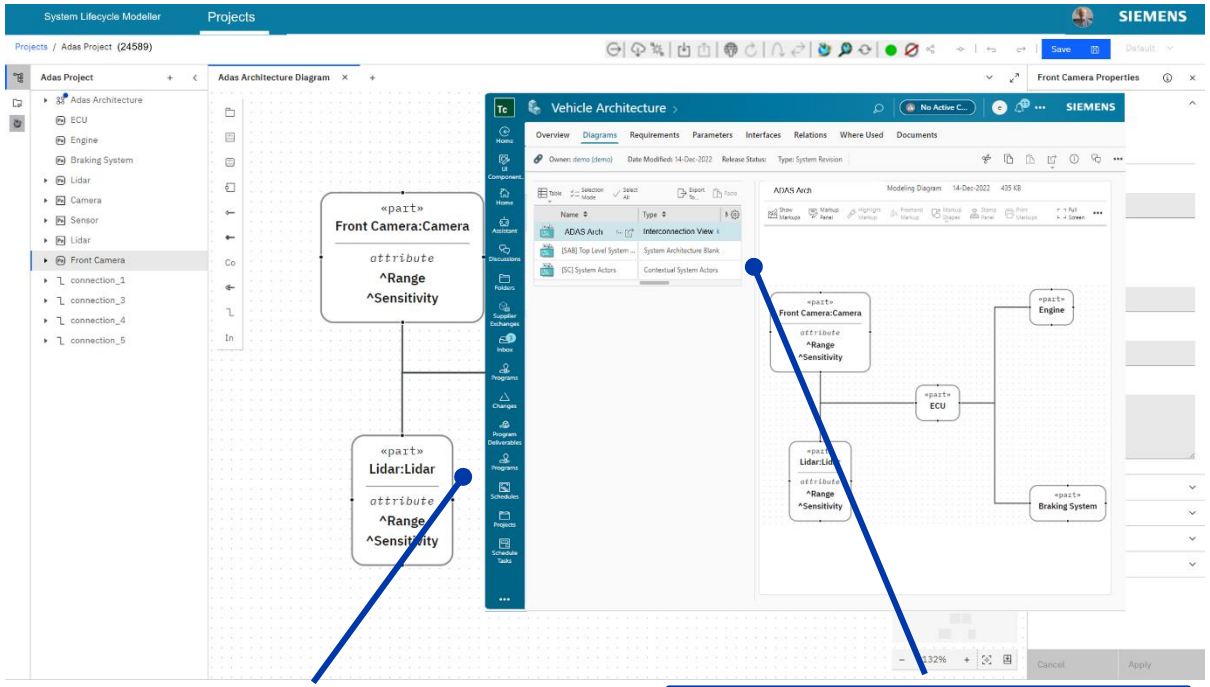
+ Extensible with plugins that Siemens Xcelerator will develop

Integration with System Lifecycle Modeler

* Stretch Goal

2412*

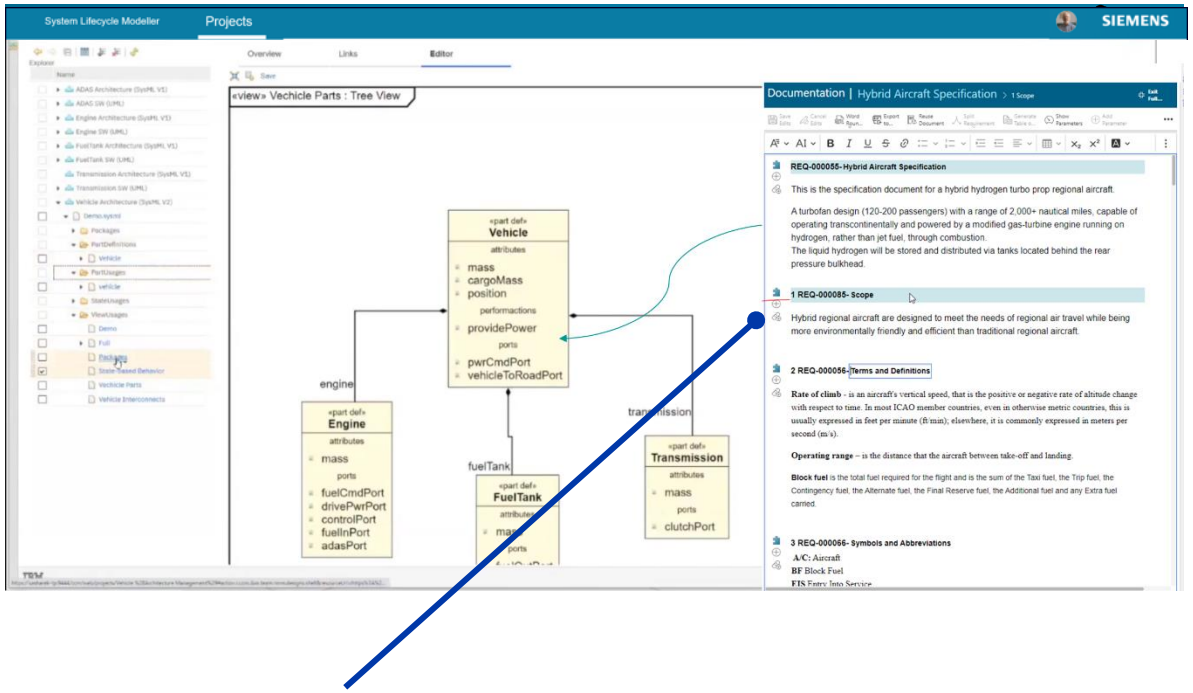
SysML V2 Authoring integrated with Teamcenter 2412



- Start integrated with your SysML V2 project lifecycle managed inside Teamcenter with known good content
- Speed up your reviews of SysML v2 product architecture with interactive navigation and views' annotation in Teamcenter

Teamcenter Requirement Integration with System Lifecycle Modeler

25xx



Teamcenter Requirement Inside

Teamcenter Inside : Requirement integration 25xx

- Tie configured requirements into your SysML V2 system for continuous requirements compliance
- Maintain and view complete cross-domain impact from requirements, to system models, to test cases, program tasks, and more, for complete traceability

So what does SysMLv2 do for me?

The diagrams are not the model!

Multiple users editing the model concurrently. Live concurrent viewing

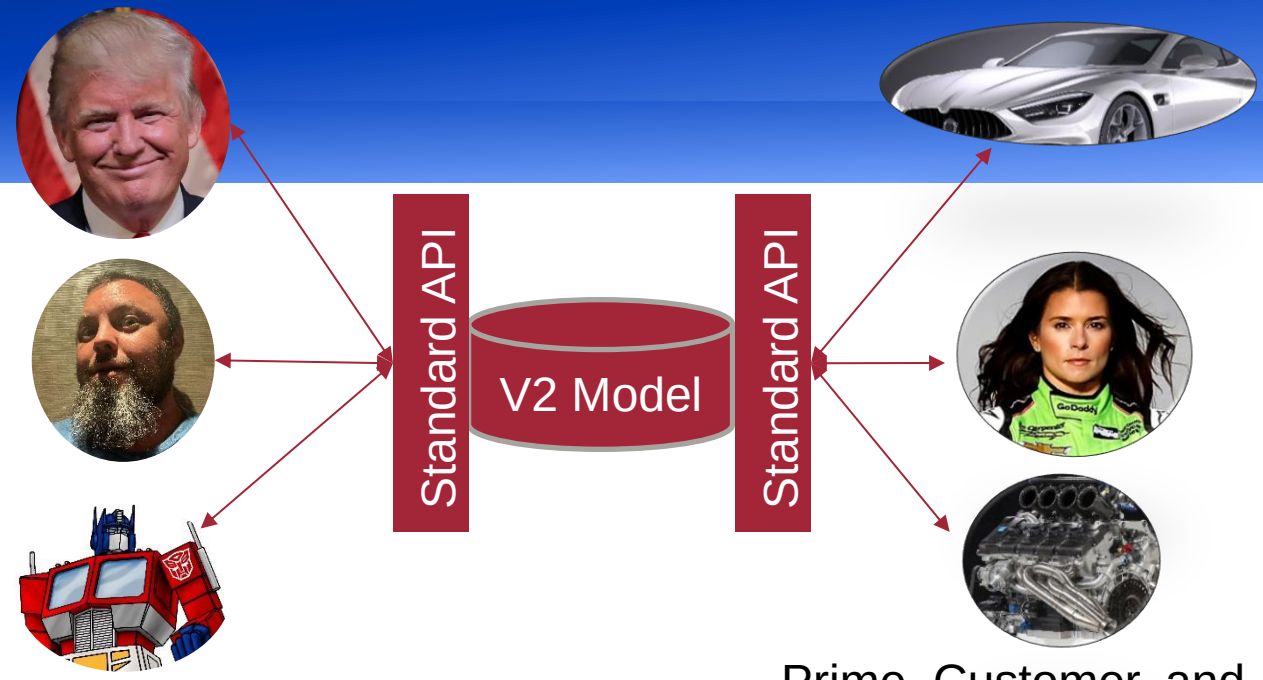
Verification, execution, analysis, rendering, of model from standard API calls and multiple vendor tools

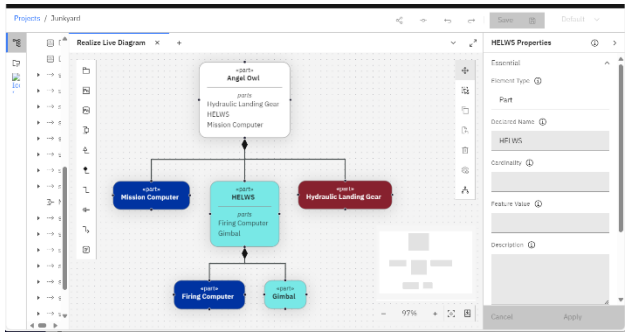
- Truly get to best of breed tool interoperability

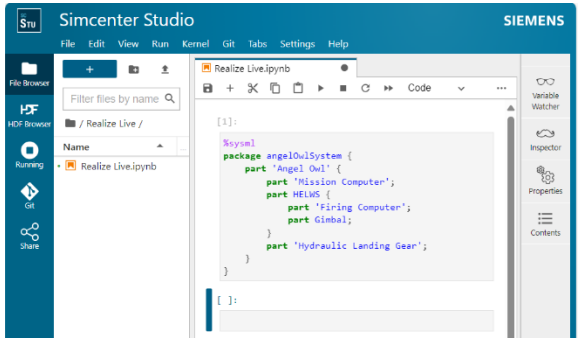
Add to the model from multiple domains

Context switch from large to small and back

Filter and view the model by concerns, including reviewers, customers and suppliers







Contact

Published by Siemens 2024

Mike Crist
Industry Strategist
Aerospace, Defense and Transport

Phone +1 903 209 8539

E-mail Michael.crist@siemens.com

Disclaimer

- **© Siemens 2024**
- **Subject to changes and errors. The information given in this document only contains general descriptions and/or performance features which may not always specifically reflect those described, or which may undergo modification in the course of further development of the products. The requested performance features are binding only when they are expressly agreed upon in the concluded contract.**
- **All product designations may be trademarks or other rights of Siemens AG, its affiliated companies or other companies whose use by third parties for their own purposes could violate the rights of the respective owner.**