MathWorks

SWISS ED25

System-on-Chip Meets Systems Engineering

A Simulation-Driven Approach

Stephan van Beek

Consulting Application Engineer
MBSE/SoC/FPGA

Christoph Kammer

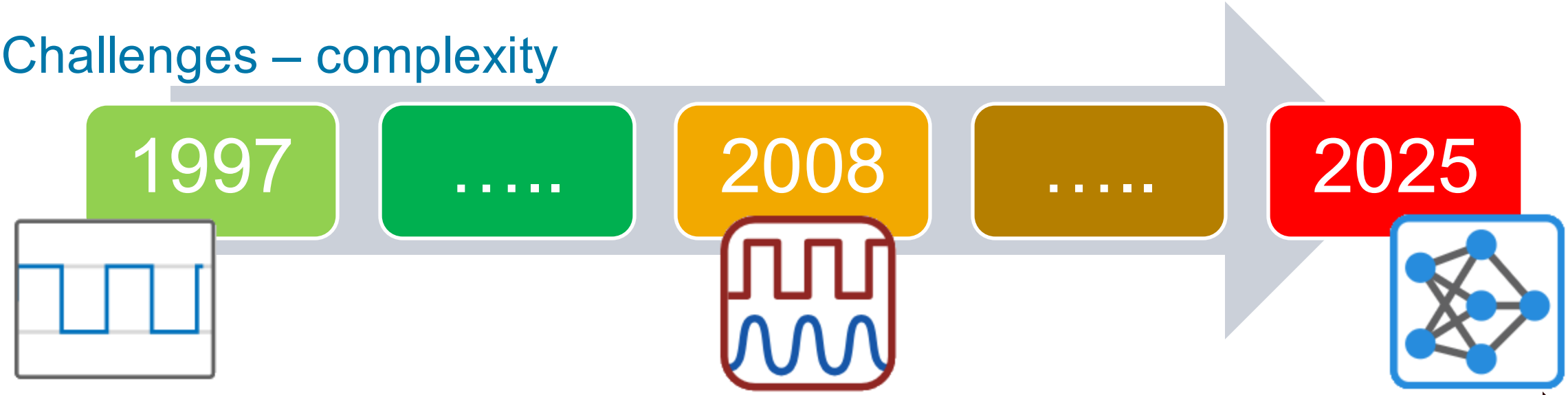
Senior Application Engineer
DevOps

Mars Climate Orbiter

Image credit: NASA/JPL-Caltech

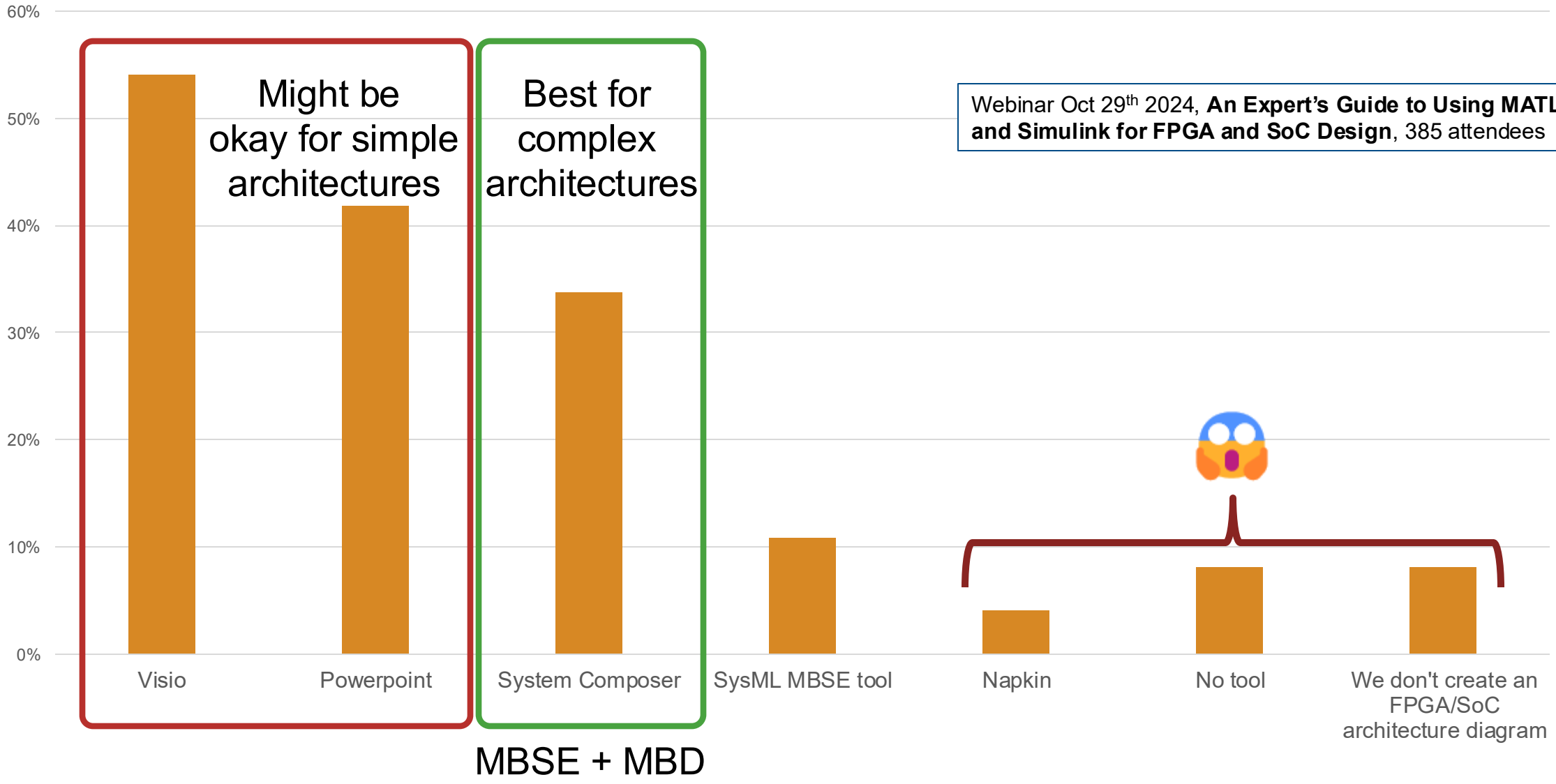
The biggest problem was not
the unit mismatch itself,
but the failure to detect and
correct this mistake

Challenges – complexity



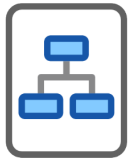
technology	FPGAs	bigger and better FPGAs	SoC = FPGA + ARM
	logic interfaces	algorithms	advanced algorithms
people	1 engineer	multiple engineers	multidisciplinary team
process	schematic	VHDL	Model-Based Design (MBD)	MBD + MBSE??

How do you create your FPGA architectural block diagrams?



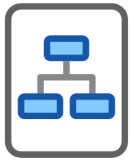
Webinar Oct 29th 2024, **An Expert's Guide to Using MATLAB and Simulink for FPGA and SoC Design**, 385 attendees

Challenges ▶ complexity
How do models and simulation help?



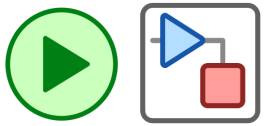
Top-down design processes

- Functional decomposition
- No simulation needed early, but ….. later you will need simulation

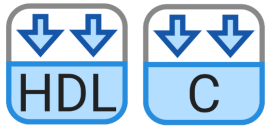


Go Beyond Textual Requirements

- Use expressiveness of requirement models and trade studies
- Use views to put stakeholder discussions in context



Validate compliance to requirements through simulation



Deployment

- Generate RTL code from **architecture + design models**

Model-Based Systems Engineering + Model-Based Design

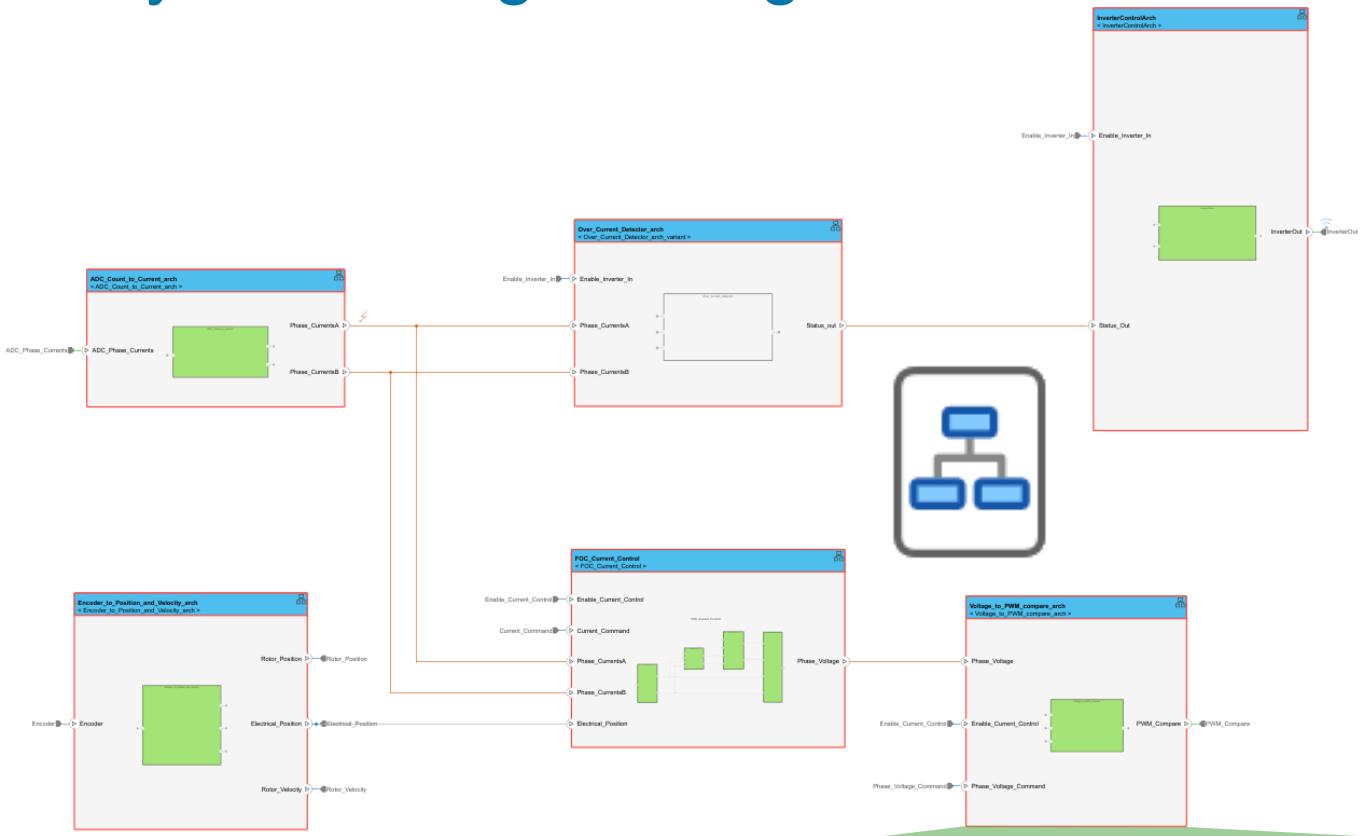
Top-down

Requirements

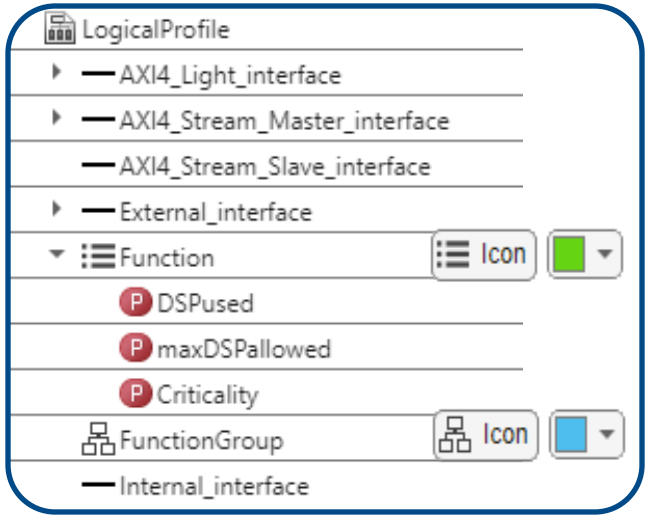
Architecture

Bottom-up

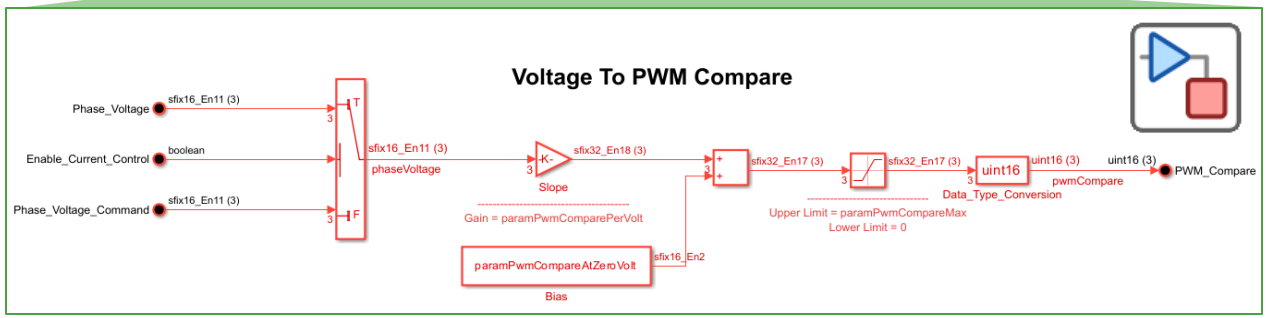
Design



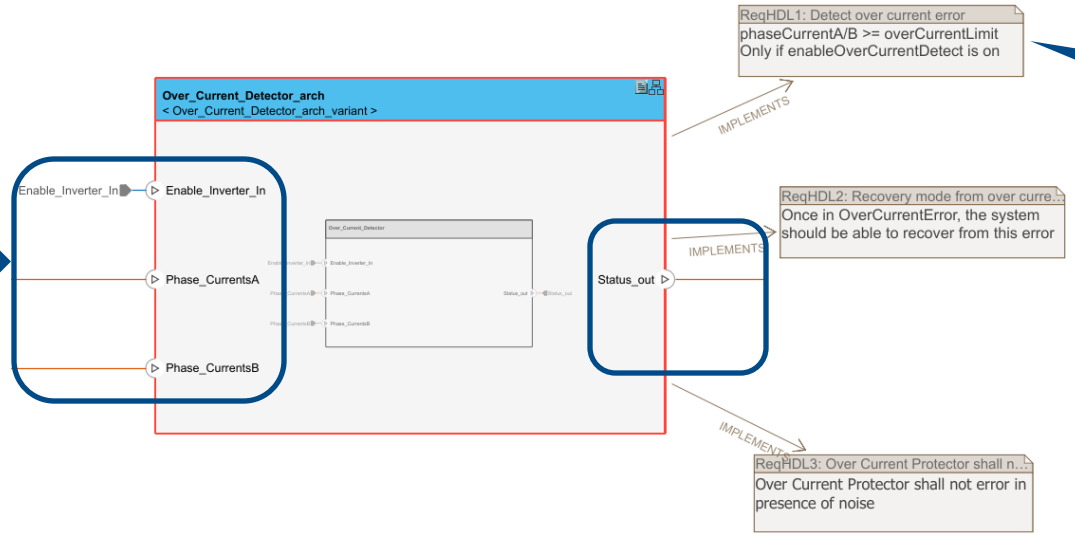
Custom Profiles



Integrate MBD with MBSE

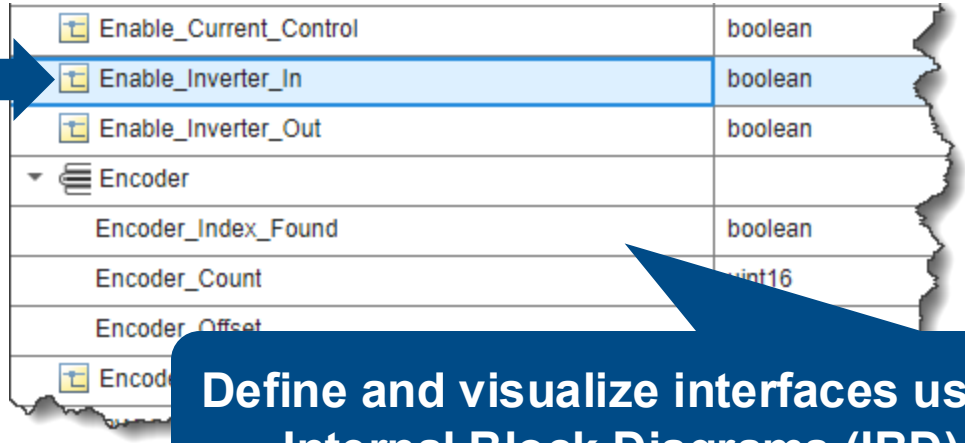


Requirements: Textual, Traceability, Interfaces, Interactions

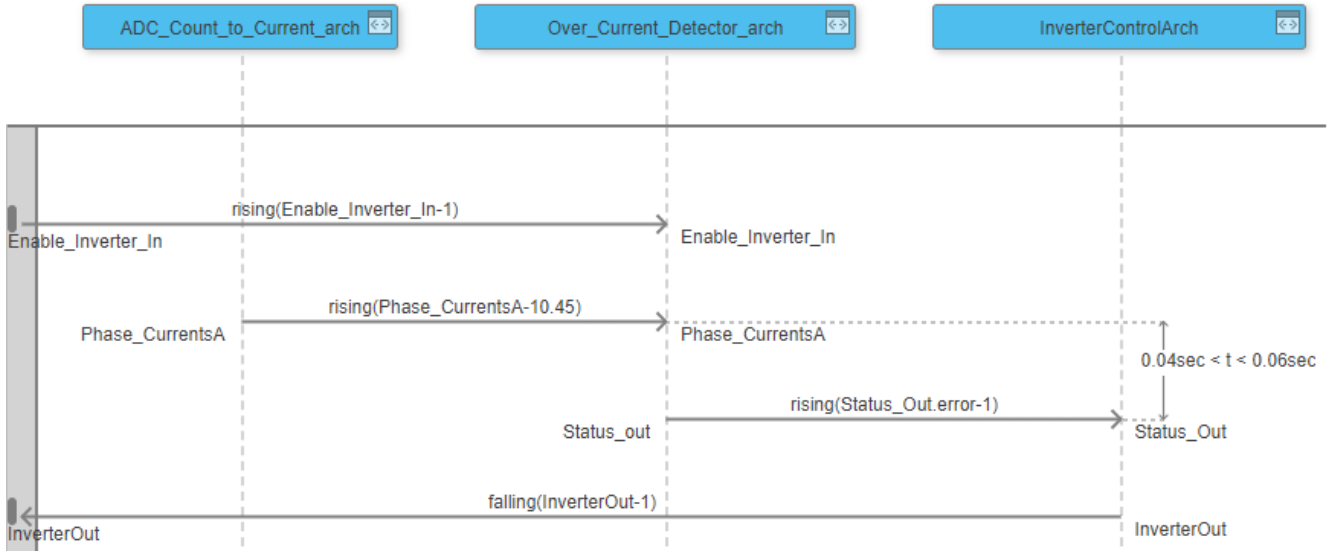


Establish traceability between architecture & design and textual requirements

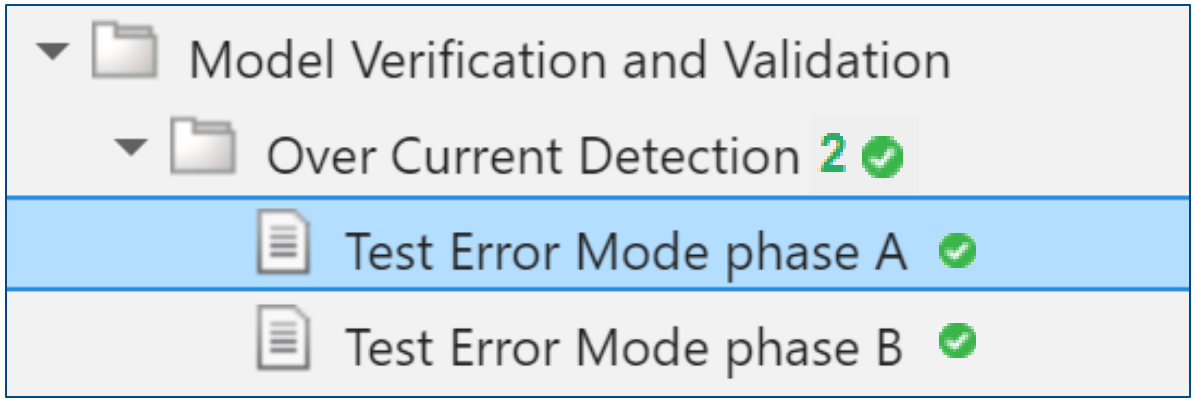
Define interface behaviors using Sequence Diagrams (SD)

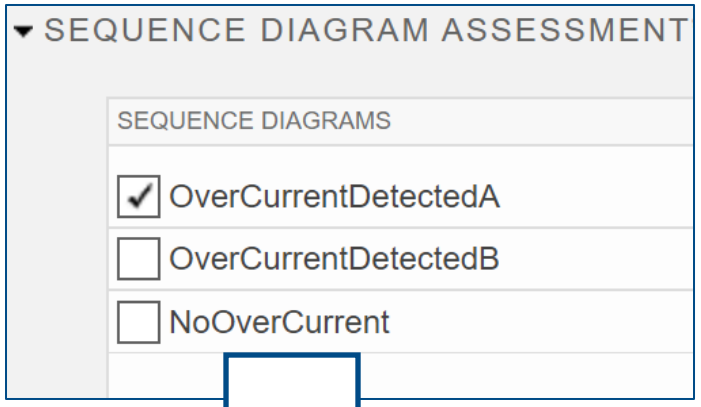


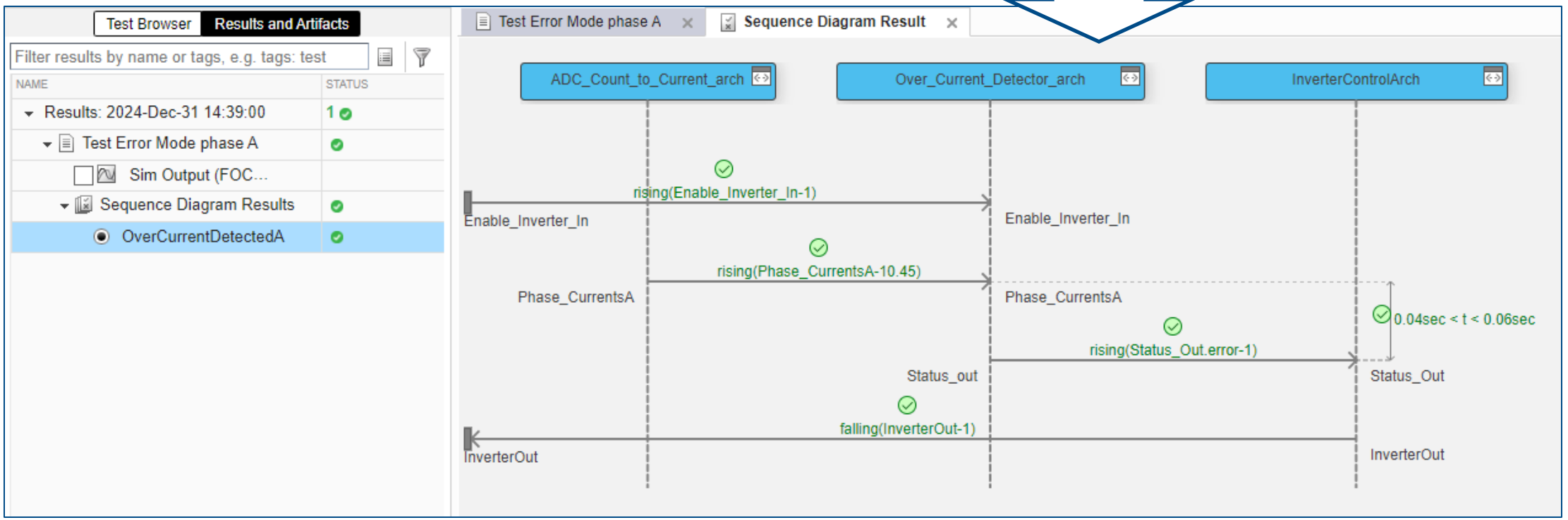
Define and visualize interfaces using Internal Block Diagrams (IBD)



Validate Systems by Re-Using Requirements Models

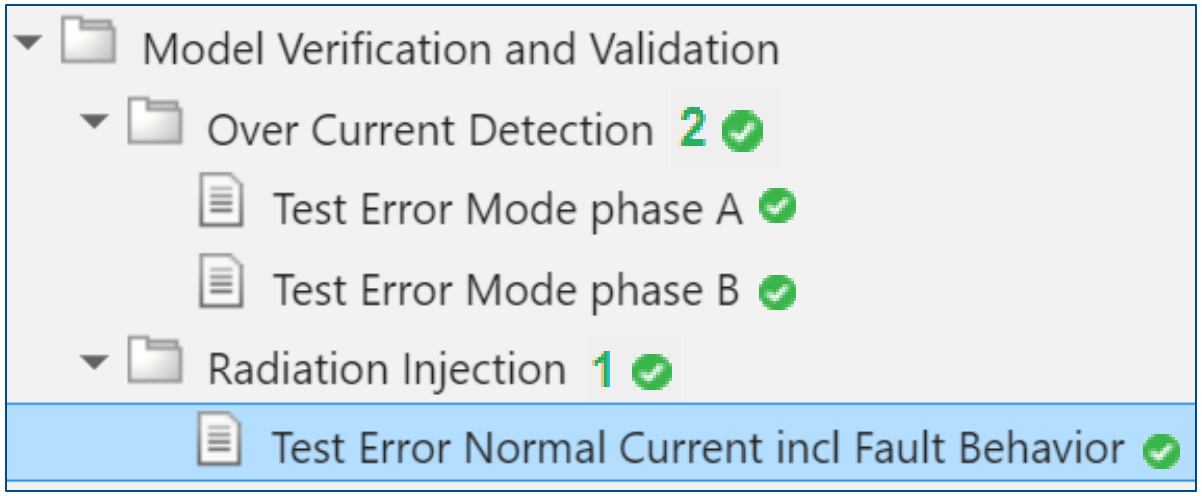


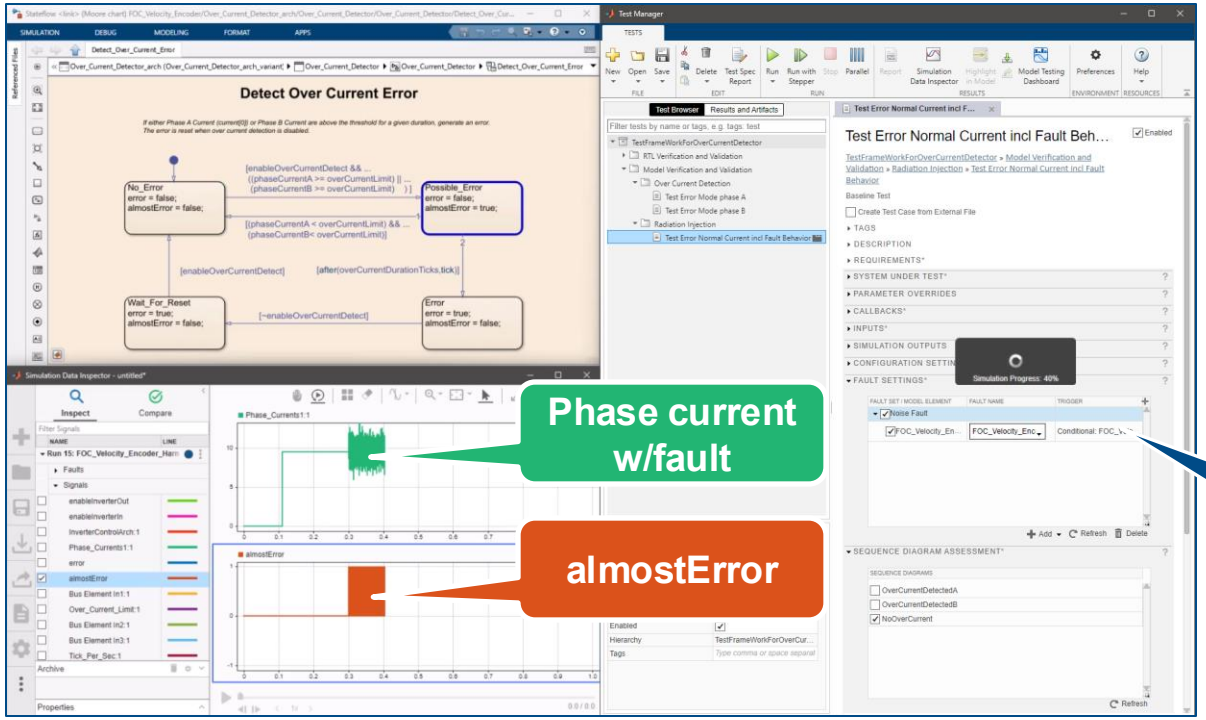


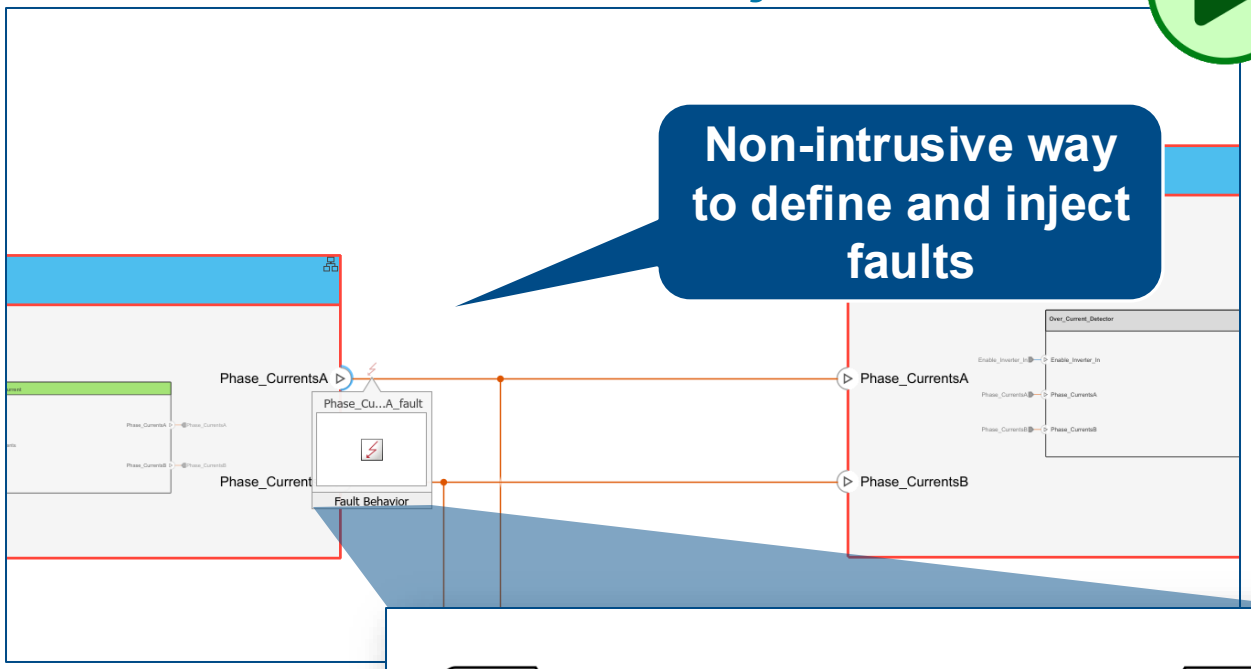


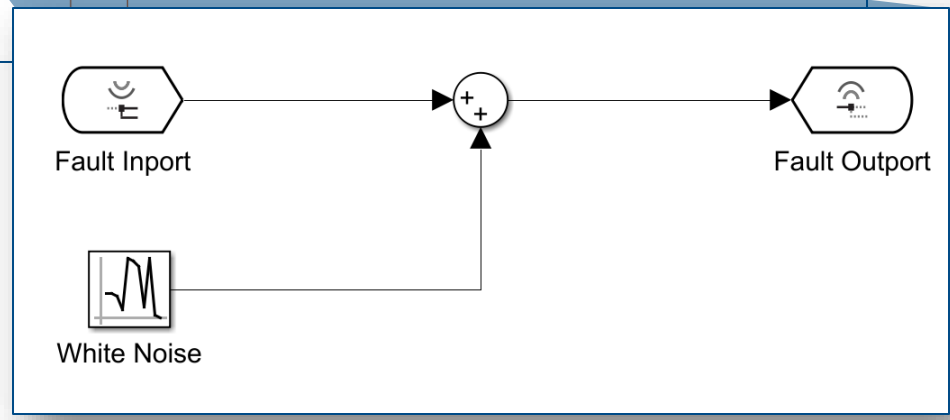

“Systems Thinking”

Validate System Behavior and Robustness with Fault Injection



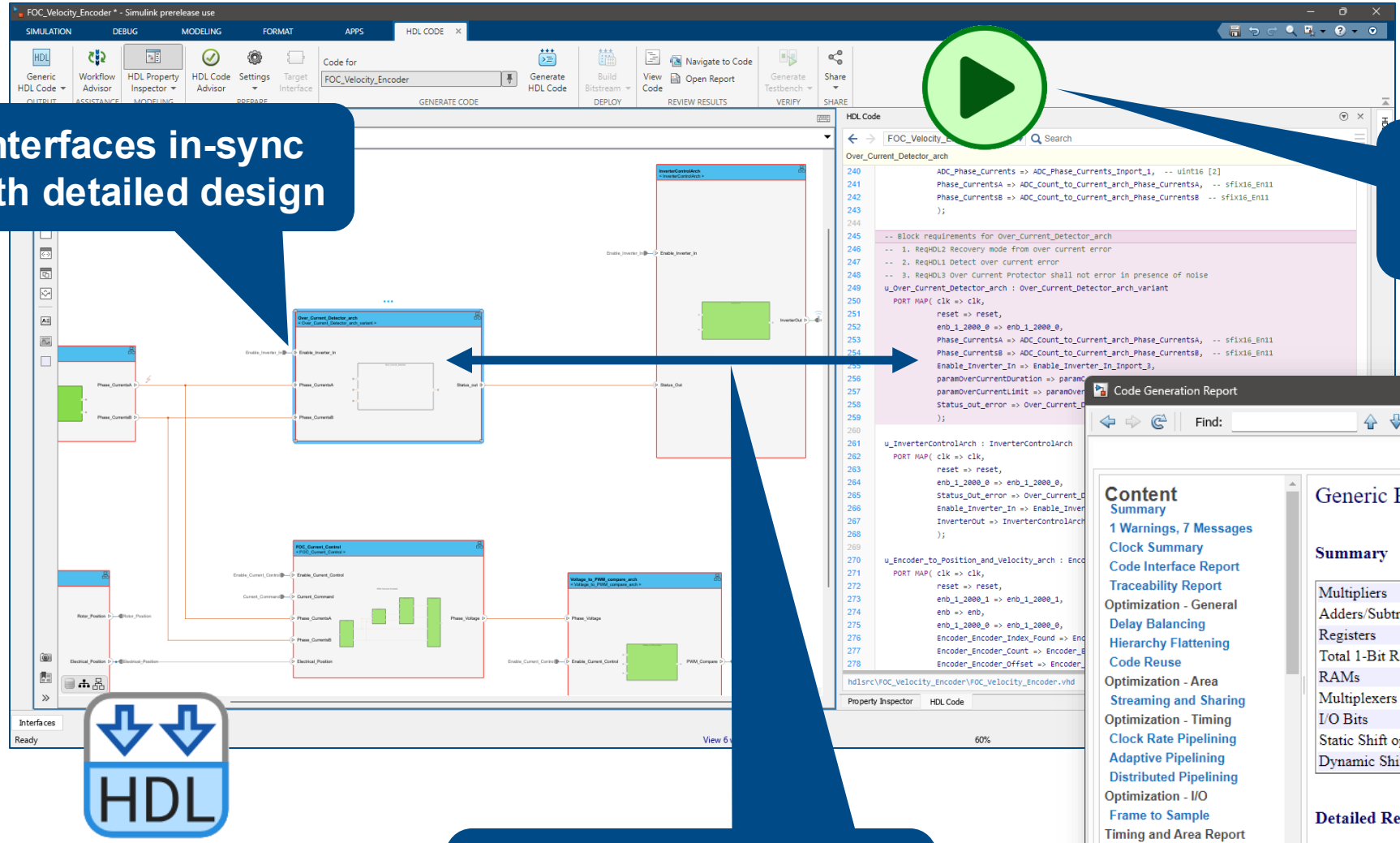






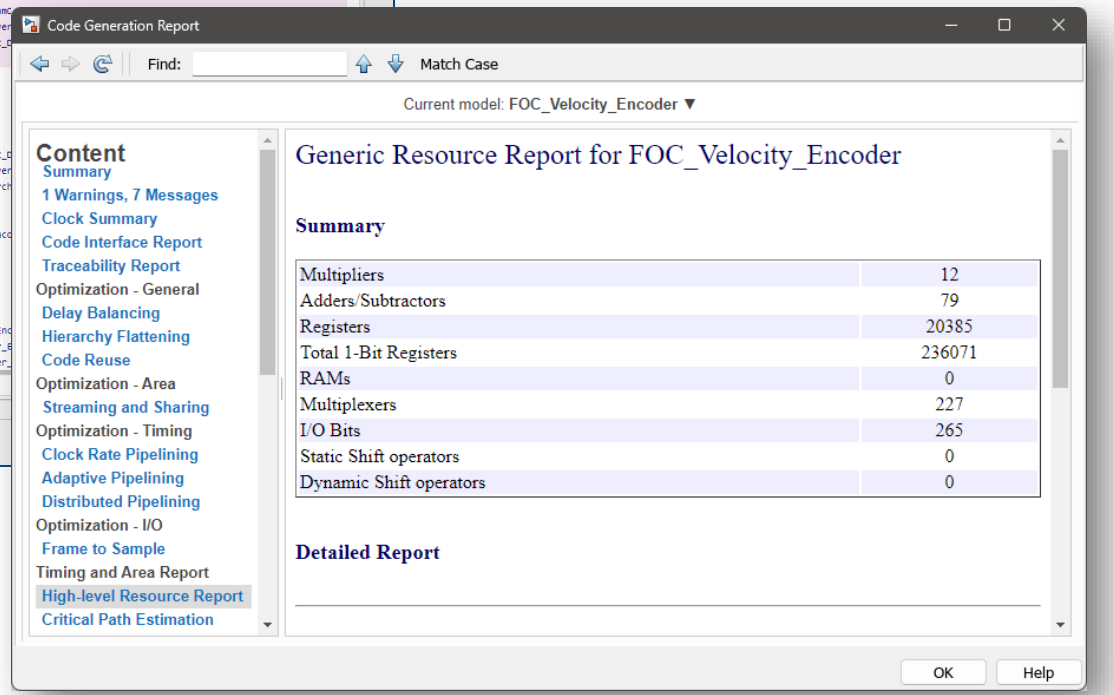
Analyze impact of faults using simulation

Generate HDL code from System Architecture incl Detailed Models



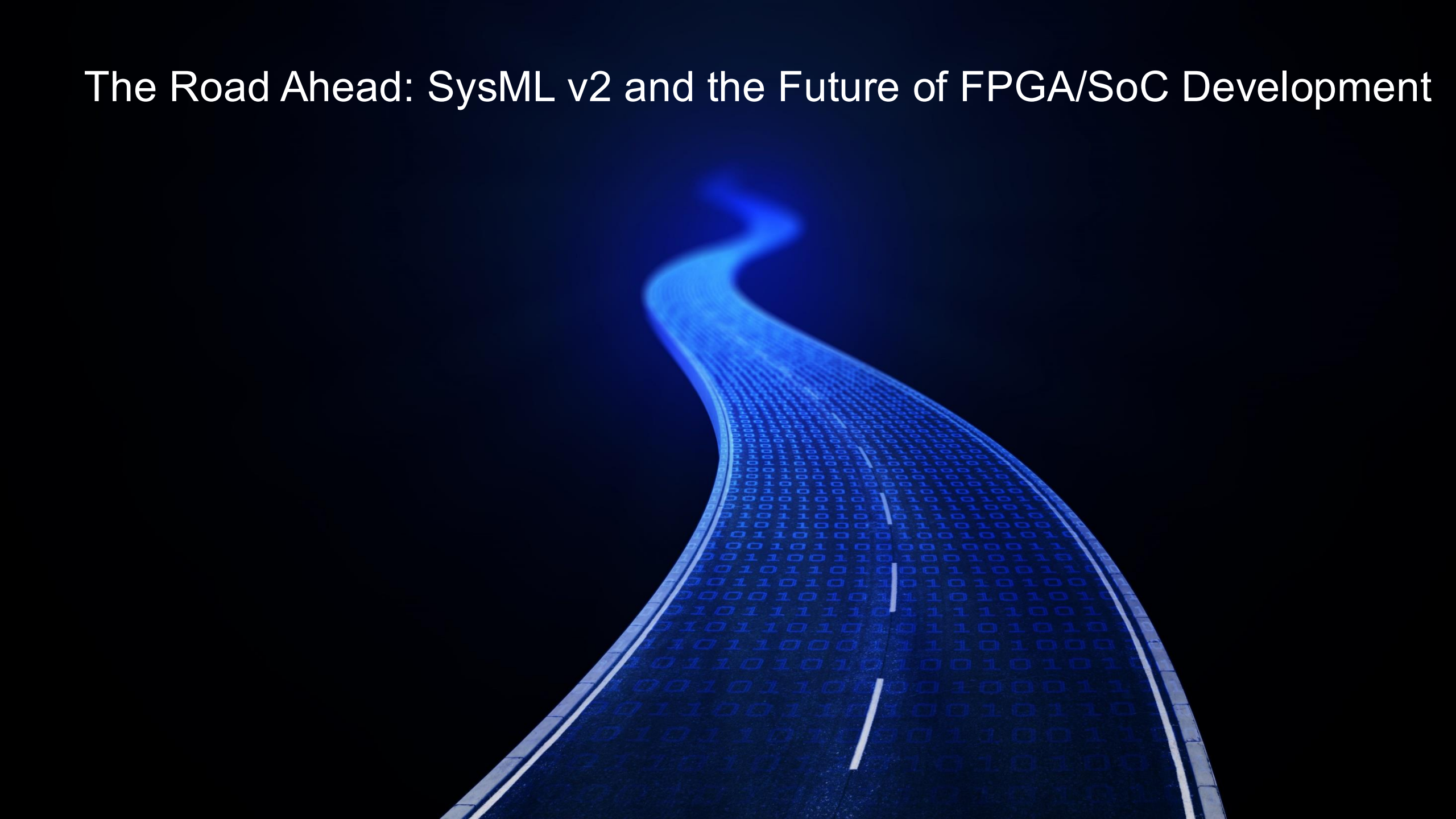
Interfaces in-sync with detailed design

Validate architecture and design before deployment

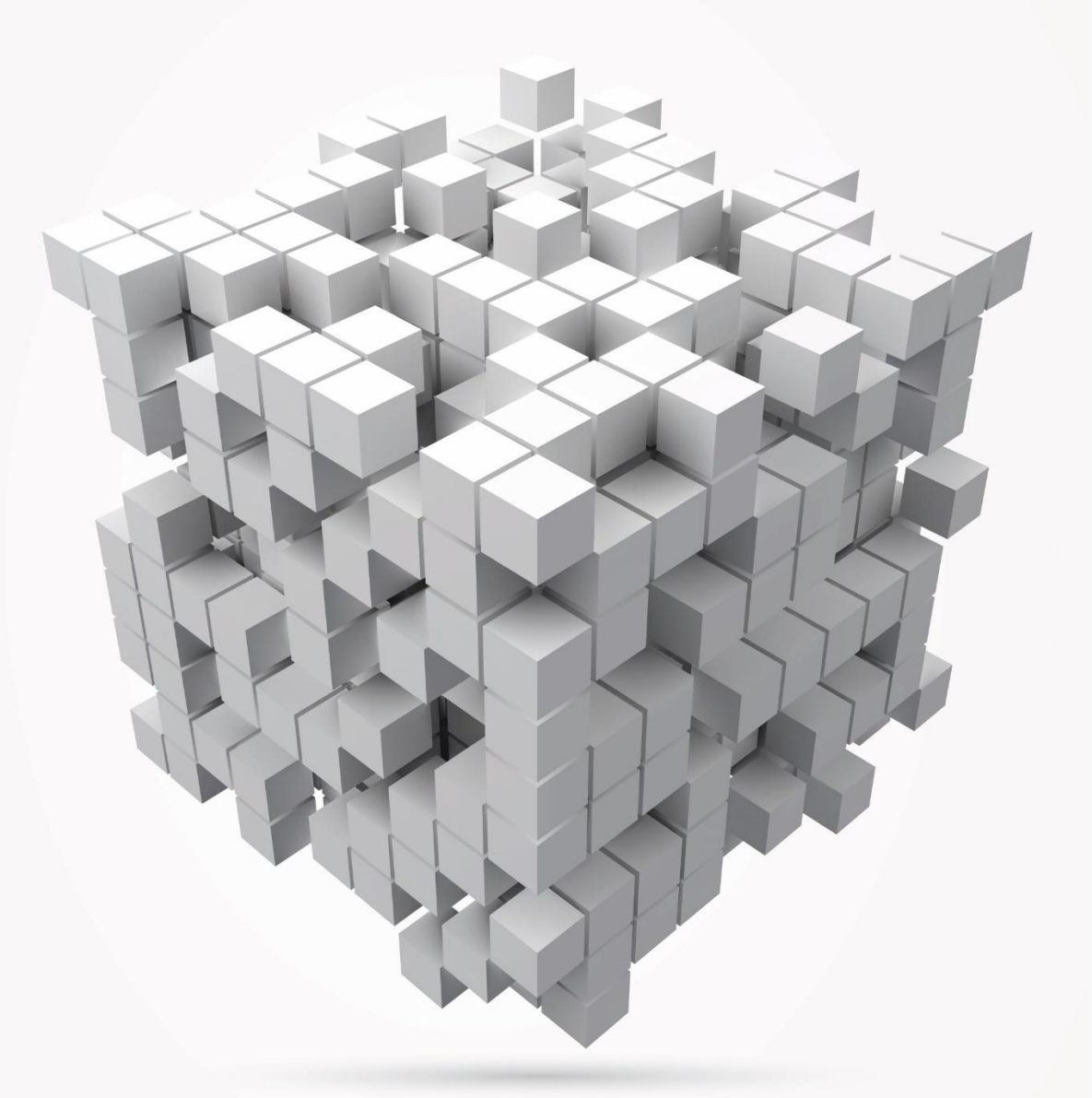


Generate traceable/ readable RTL code

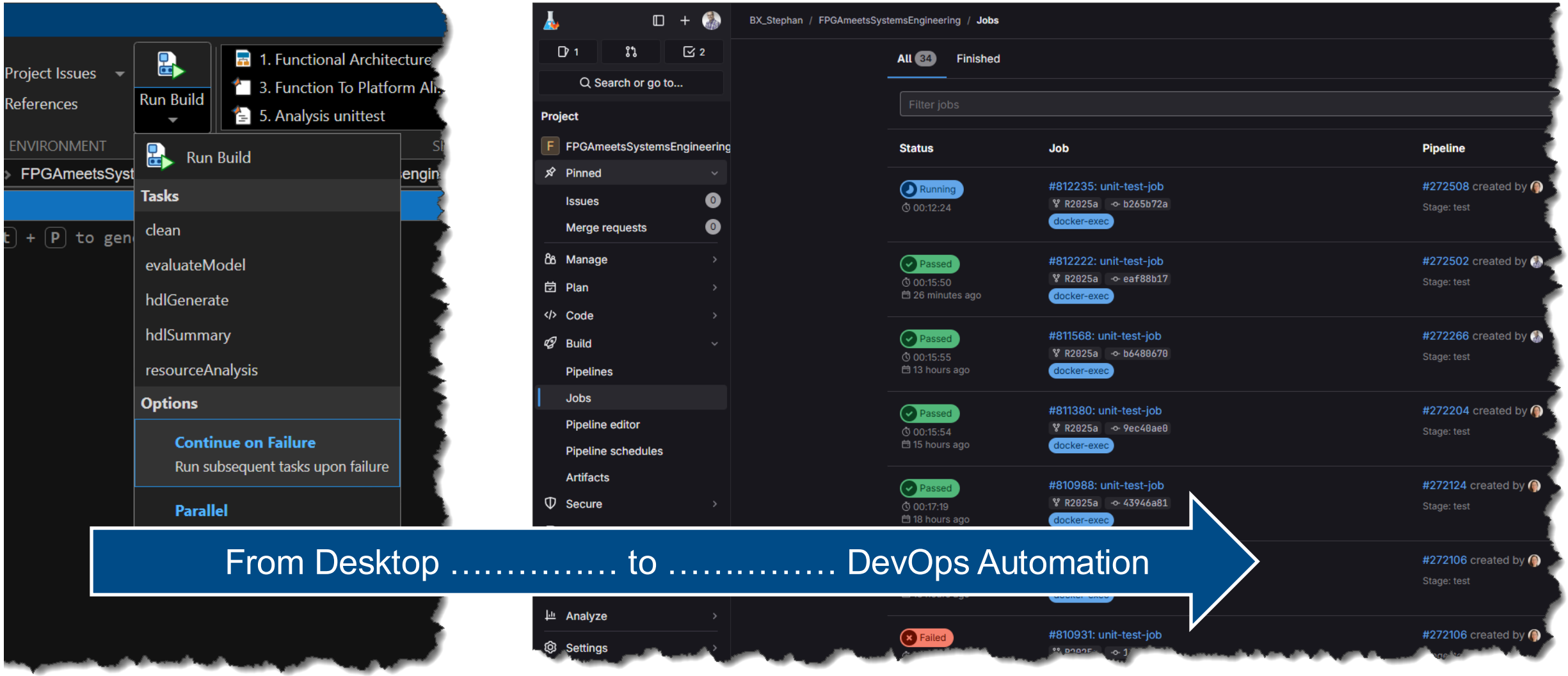
The Road Ahead: SysML v2 and the Future of FPGA/SoC Development



Automation and Interoperability

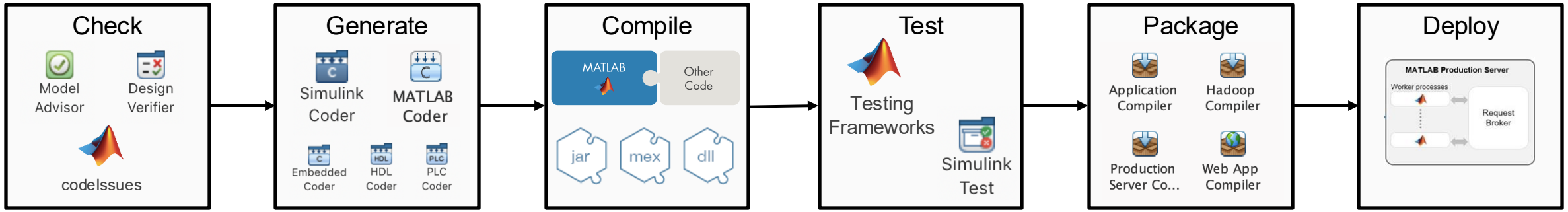


DevOps Automation to Ensure Consistent Build Processes



Automate Locally or in CI with Buildtool

- MATLAB has a growing list of capabilities that necessitate build tasks, resulting in more ad hoc scripts



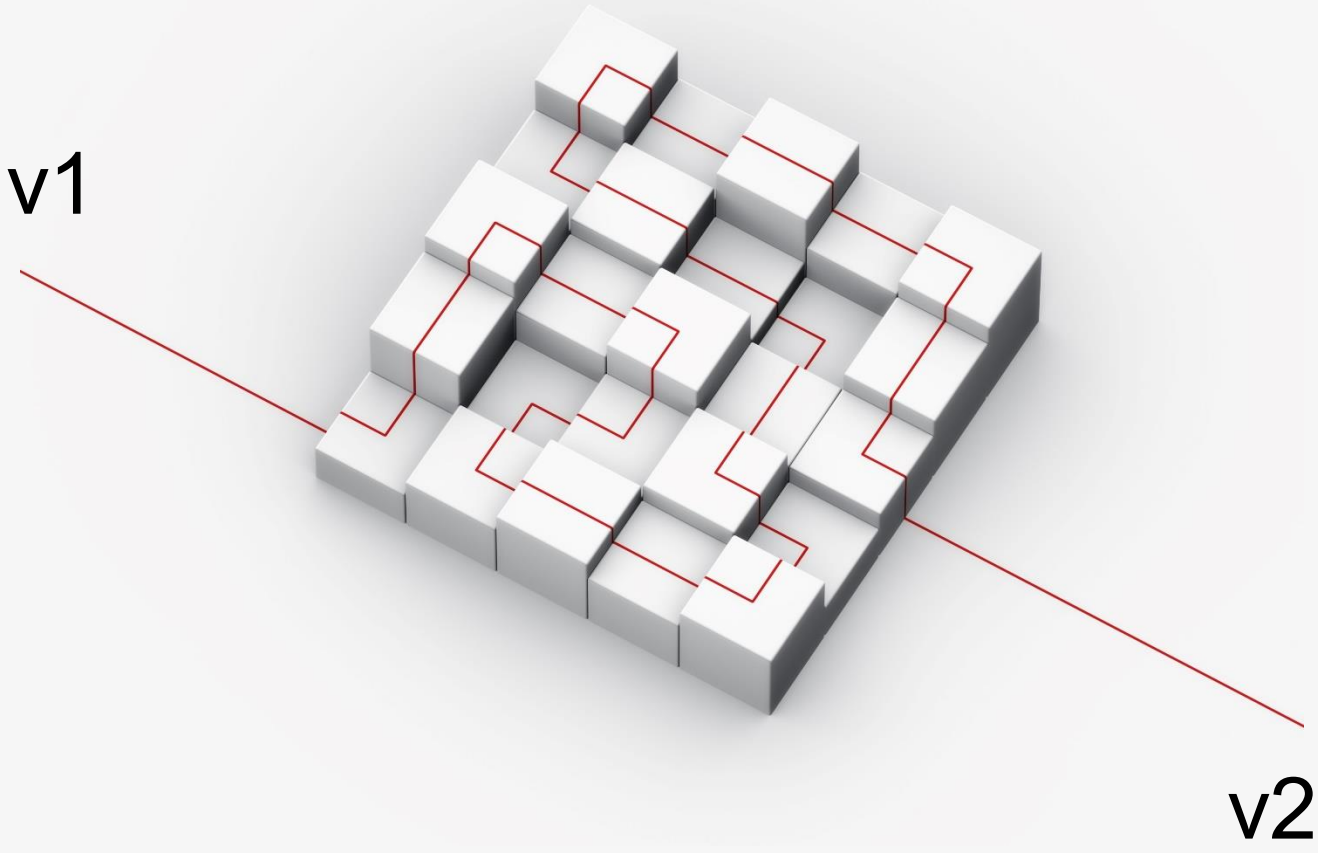
- The MATLAB Build Tool is a build system that provides a standard programming interface to create and run tasks in a uniform and efficient way

Continuous Integration, Delivery and Deployment (CD)

	Local development		CI = Continuous Integration		CD = Continuous Delivery	CD = Continuous Deployment
	Develop & test models	**Check in**	**Test**	**Build & Test**	**Release**	**Deploy to production**
Description	Author the models, test locally	Check-in changes to the source control system	Run tests periodically e.g., with every check-in	Generating HDL and C code	Use FPGA vendor tools for bitstream generation	Download bitstream & SW executable and deploy to System-on-Chip
Tools	 **MATLAB Simulink System Composer**	**GitLab**	**HDL & C Coders GitLab Docker agents**		**AMD Vivado ARM Compiler**	

Continuous Integration: CI/CD Automation for Model-Based Design (support package)
CI/CD Automation for Simulink Check (reference book)

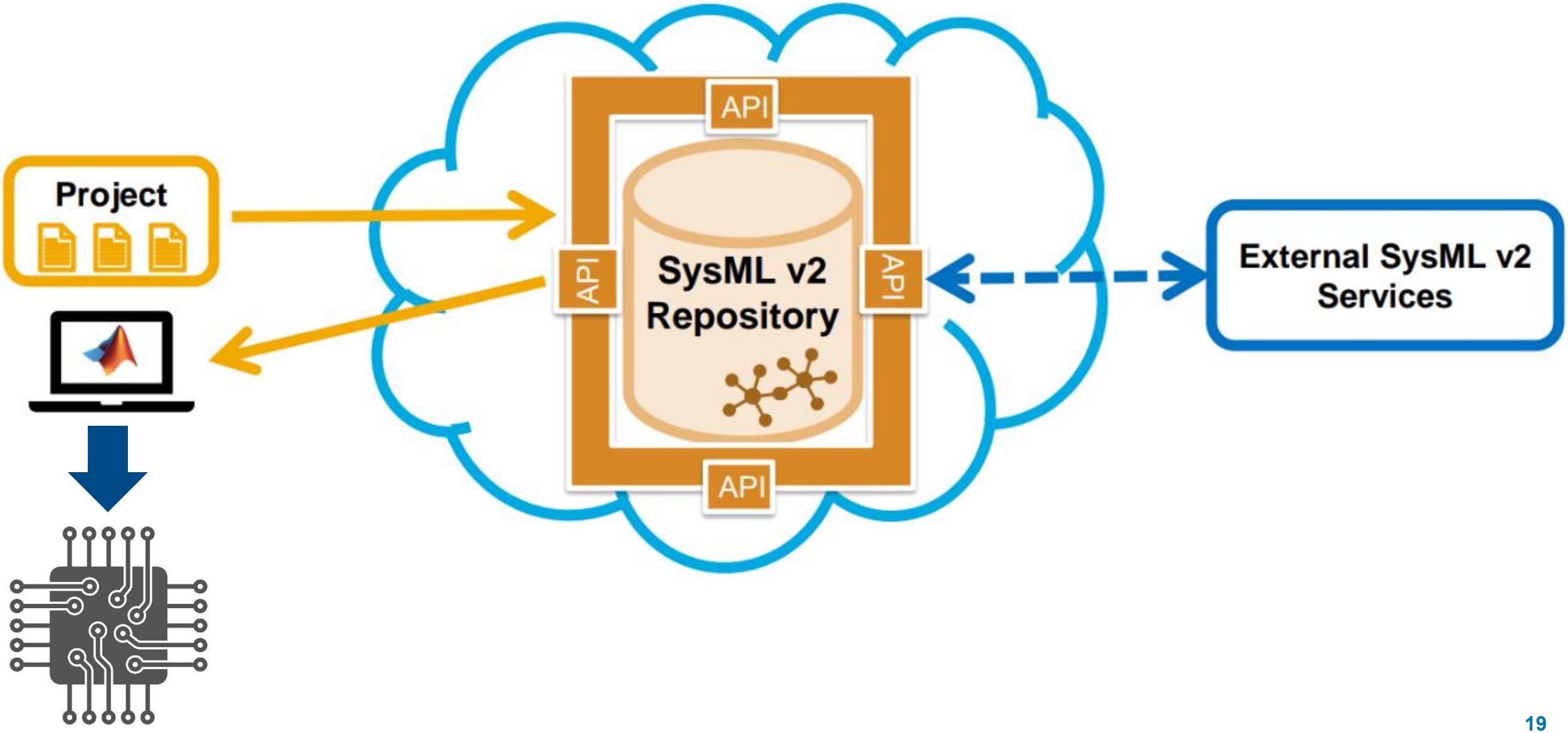
What is SysML?



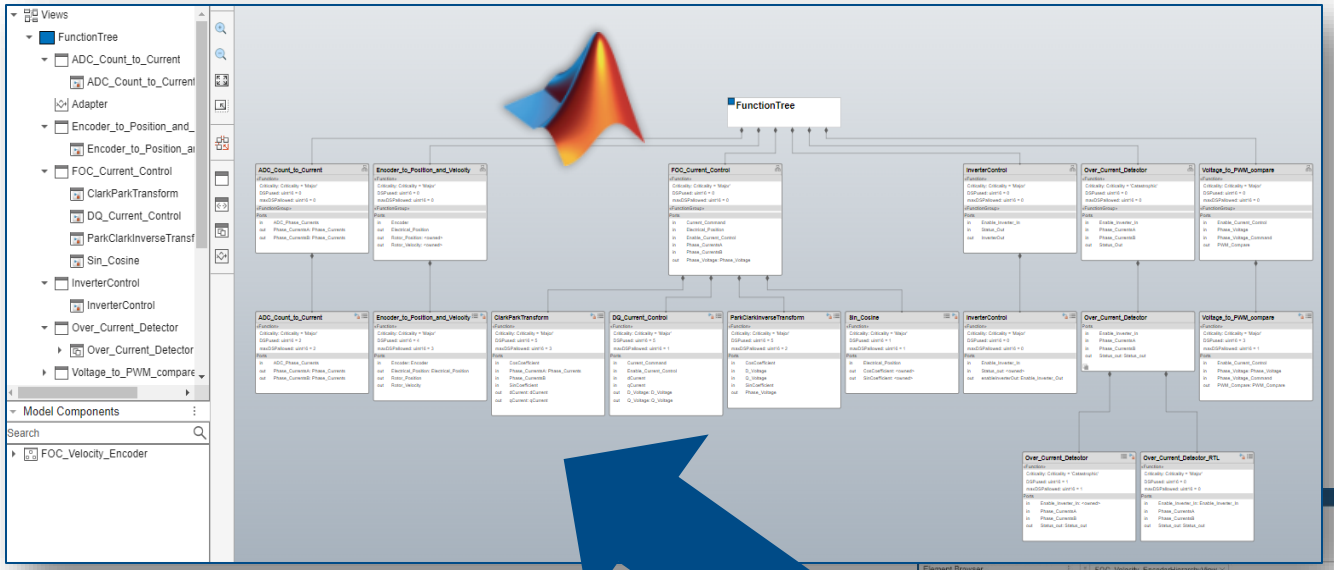
SysML is used for:

- Modeling complex systems
- Visualizing system architecture
- Describing system behavior
- Verifying system requirements

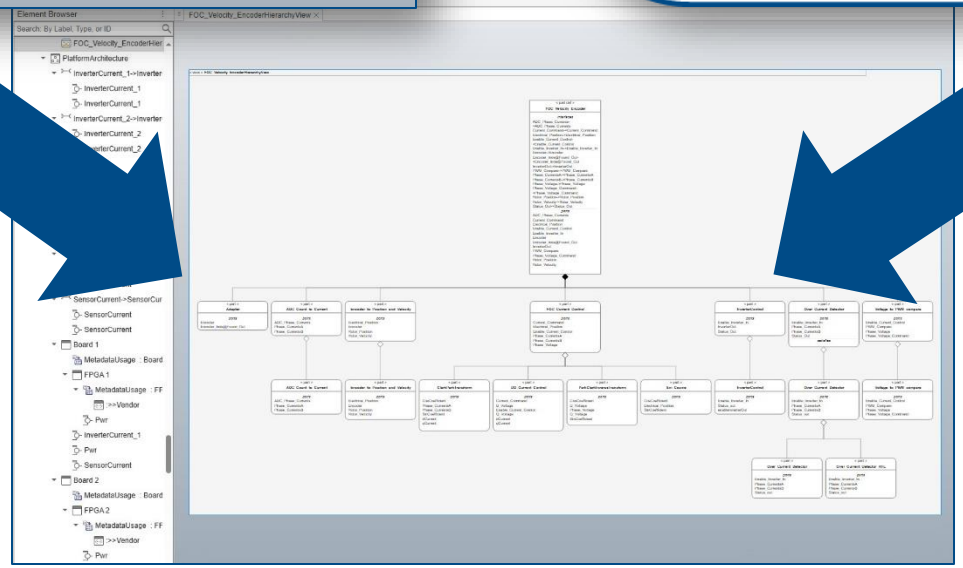
SysMLv2 and Enhanced Interoperability



Interoperability in Action



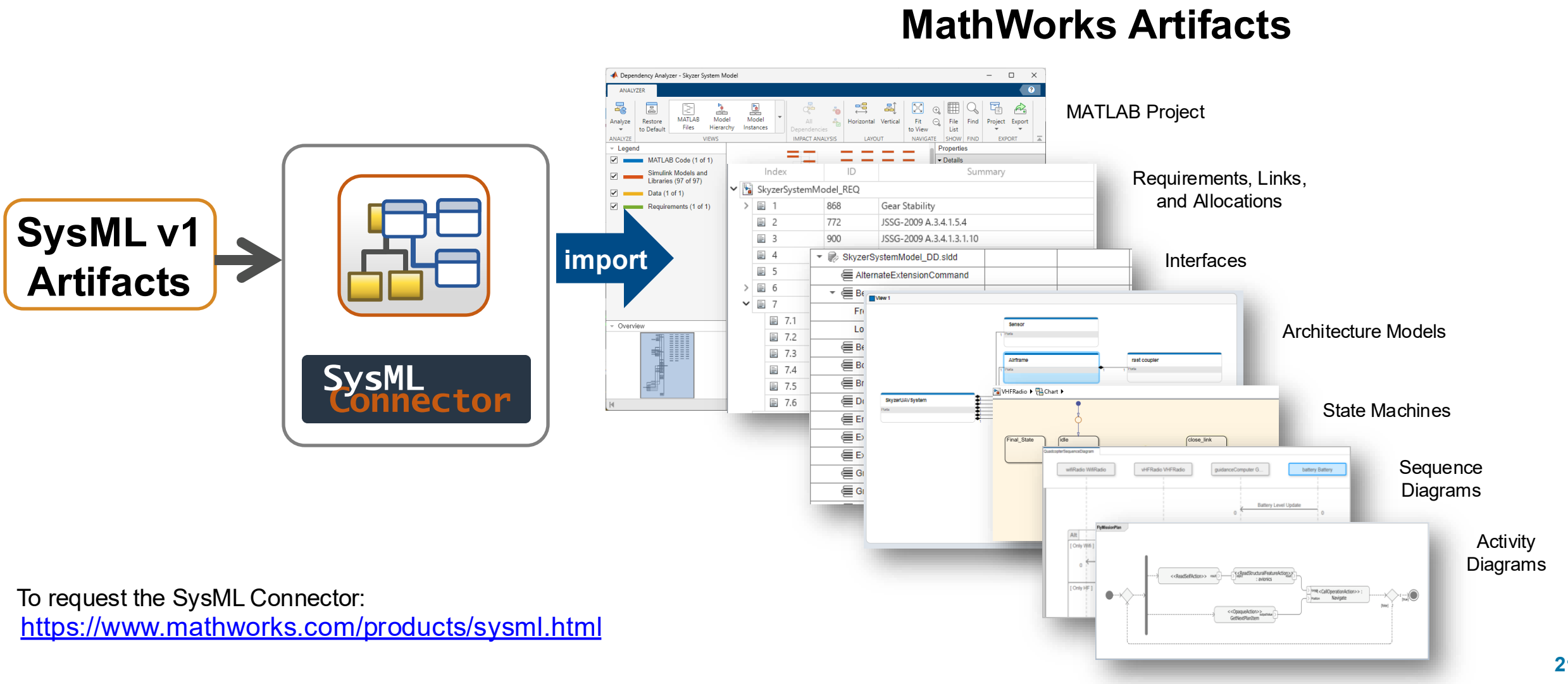
Other MBSE environments



SysMLv2 repository

Where do I start with SysMLv1?

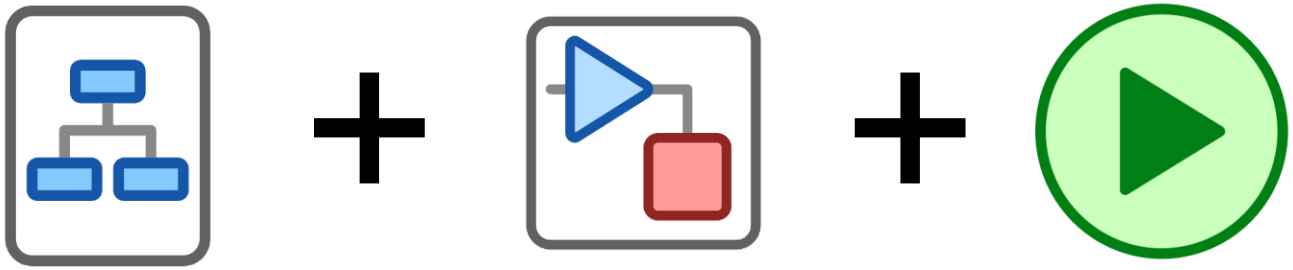
Use the SysML Connector to import your SysMLv1 model into MATLAB



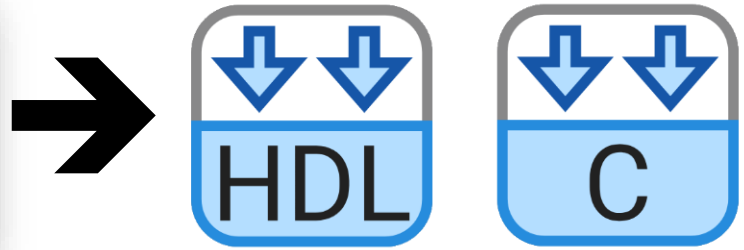
To request the SysML Connector:
https://www.mathworks.com/products/sysml.html

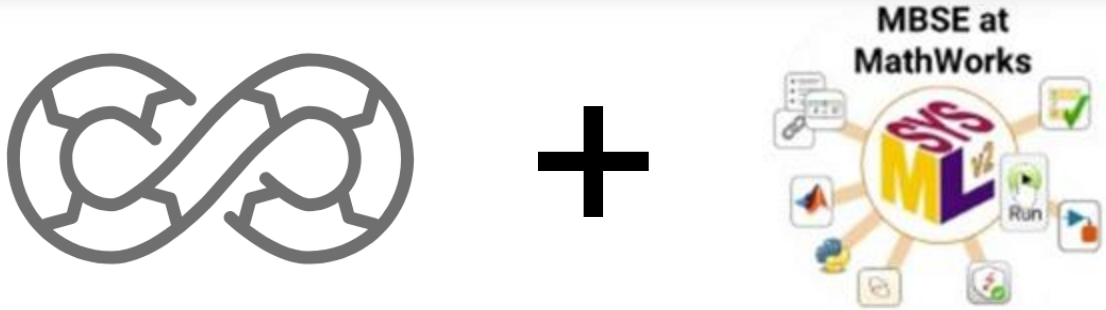
Concluding remarks

Architecture + Design + Simulation



The main FPGA challenge lies in architecture, often informal and overlooked. A tool is needed to generate RTL directly from the architecture + design in a single model, making it the single source of truth.
- Adam Taylor, Adiuvo **-- well known FPGA/SoC influencer and blogger**





Automation + Interoperability

Questions?

More information