MATLAB EXPO

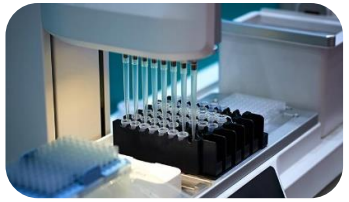
FRANCE

Concevoir, analyser et tester les architectures système

Ibrahim Saddoug, MathWorks

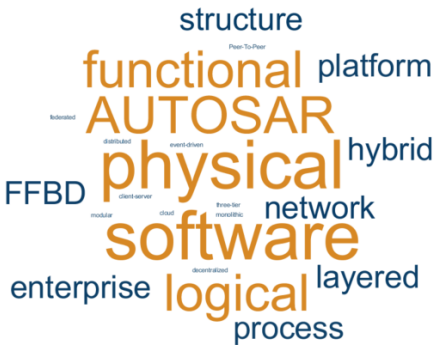
MathWorks

Design, analyze, and test system architectures

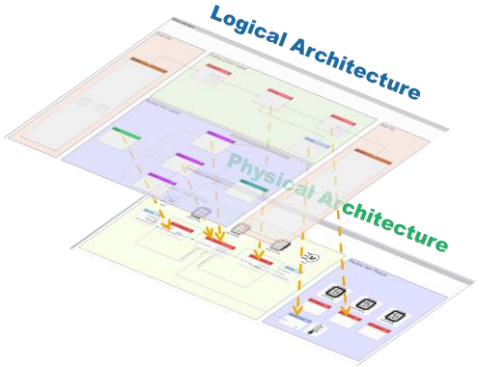
- System engineers ensure the system meets customer needs and that the design engineers do not over or under design the delivered system

- **System Composer**…
  - enables intuitive, scalable and adaptive modeling of **requirements traceability**, **architecture modeling**, **system analysis**, and more, **EARLY** in the development process
  - enables trade studies to quantify decision making in conjunction with engineering judgement **EARLY** in the development process before Simulink even is involved
  - directly connects with Simulink's Model Based Design
  - enables customer's system engineering processes to be done in the same toolset as their design allowing reuse of **EARLY** work done

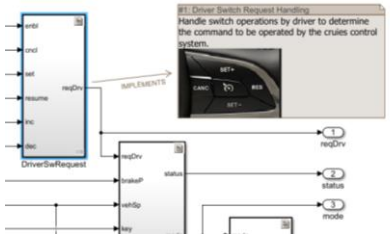


Key Takeaways

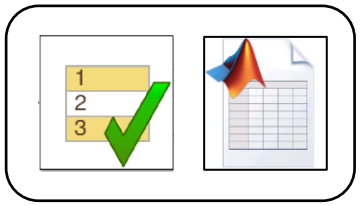
Capture and manage stakeholder's **needs** and describe system **architectures**



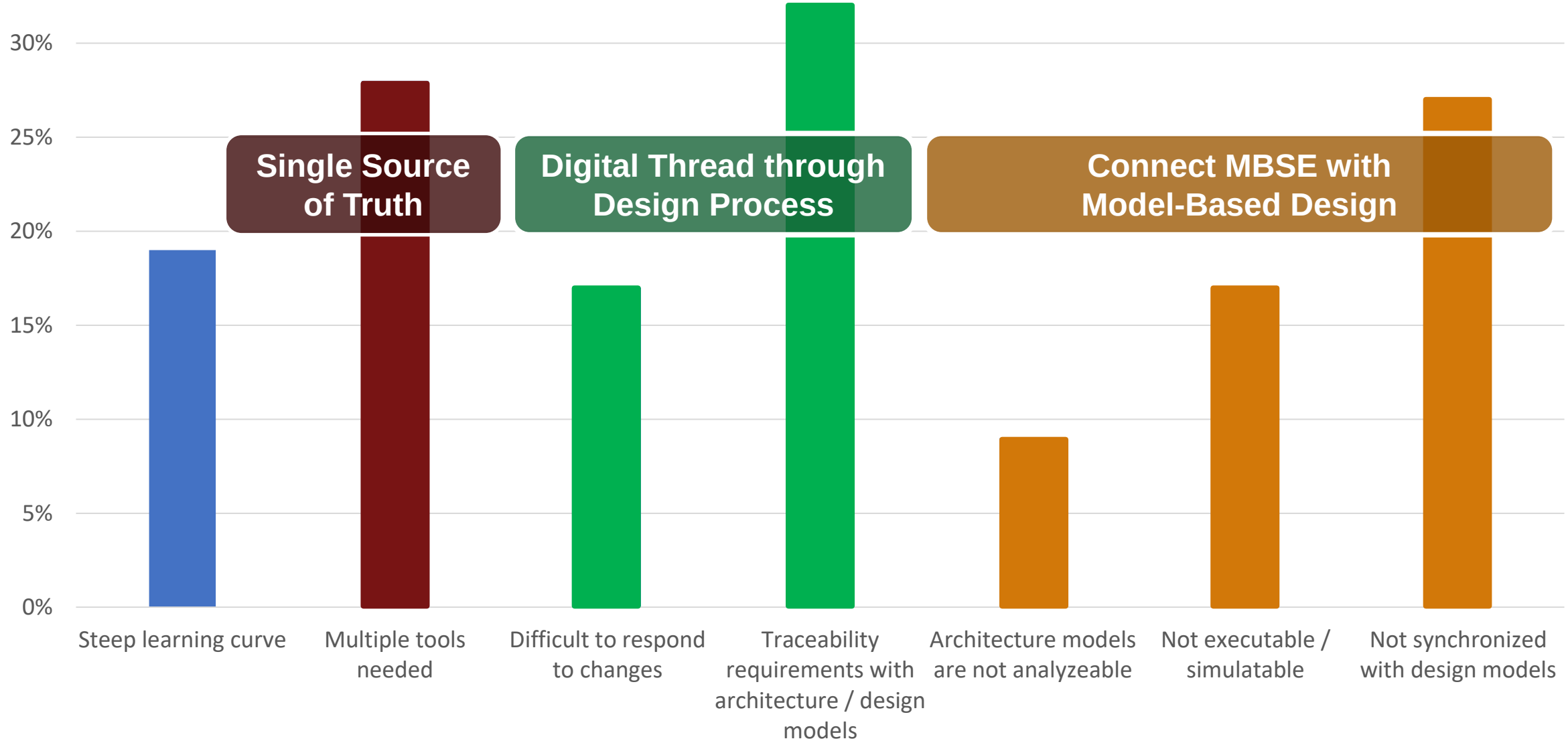
Linking requirements, system architectures, simulation models, and perform **trade-off analysis**



Achieve verification and validation through **simulation**

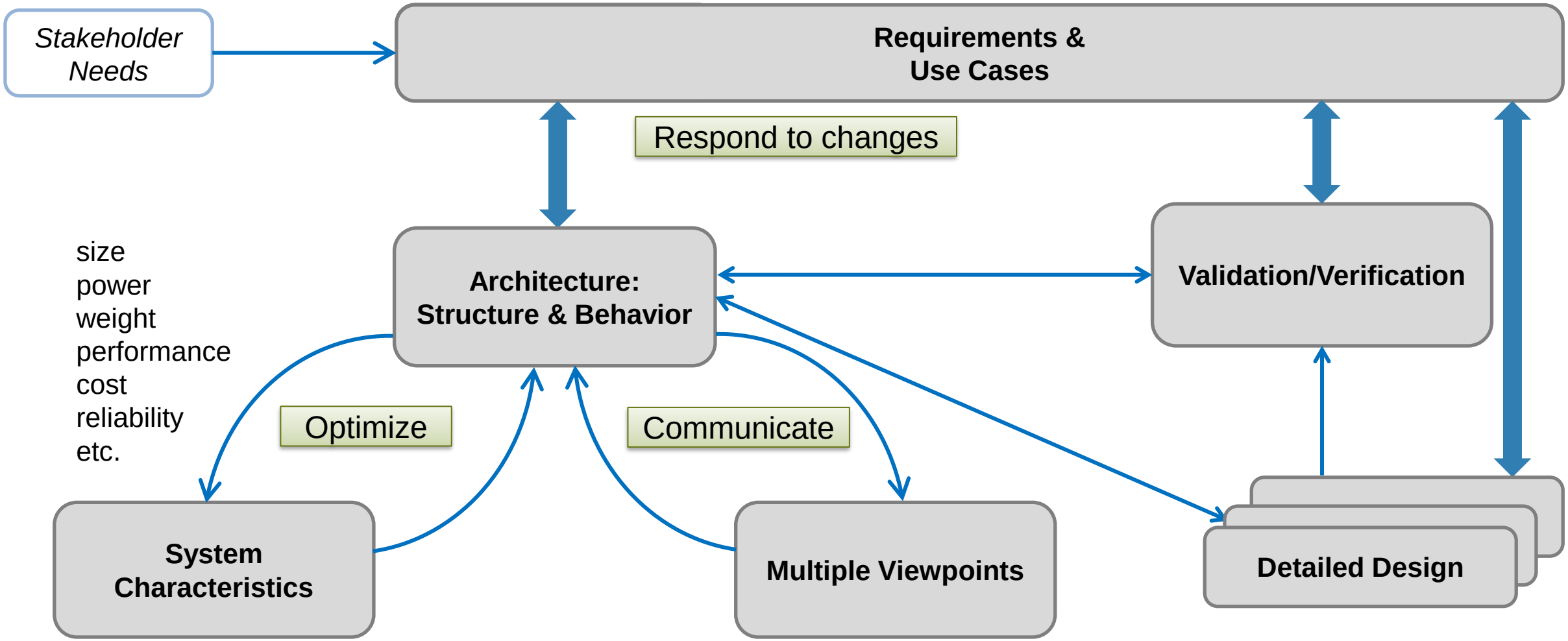


What makes Systems Engineering challenging?

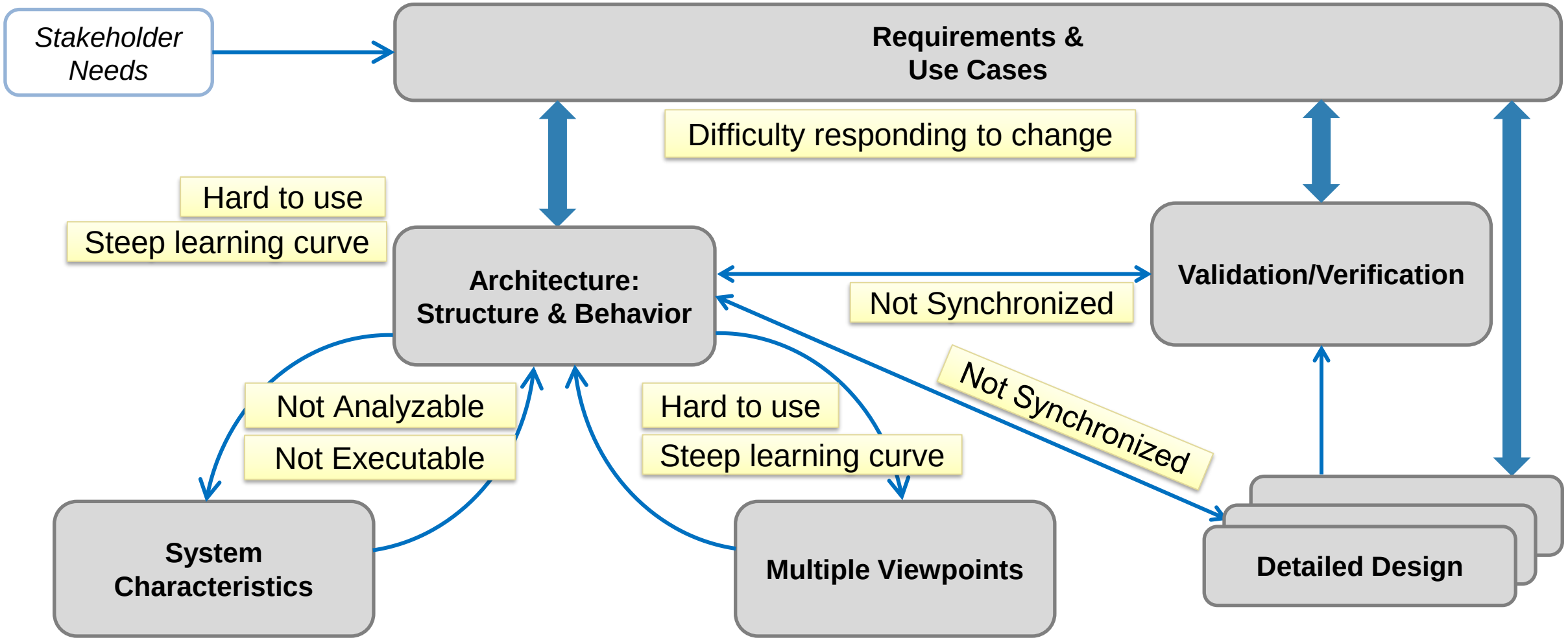


Source: MathWorks webinar, Model-Based Systems Engineering - Practical Use and Applications, December 9th, 2021 (173 responses)

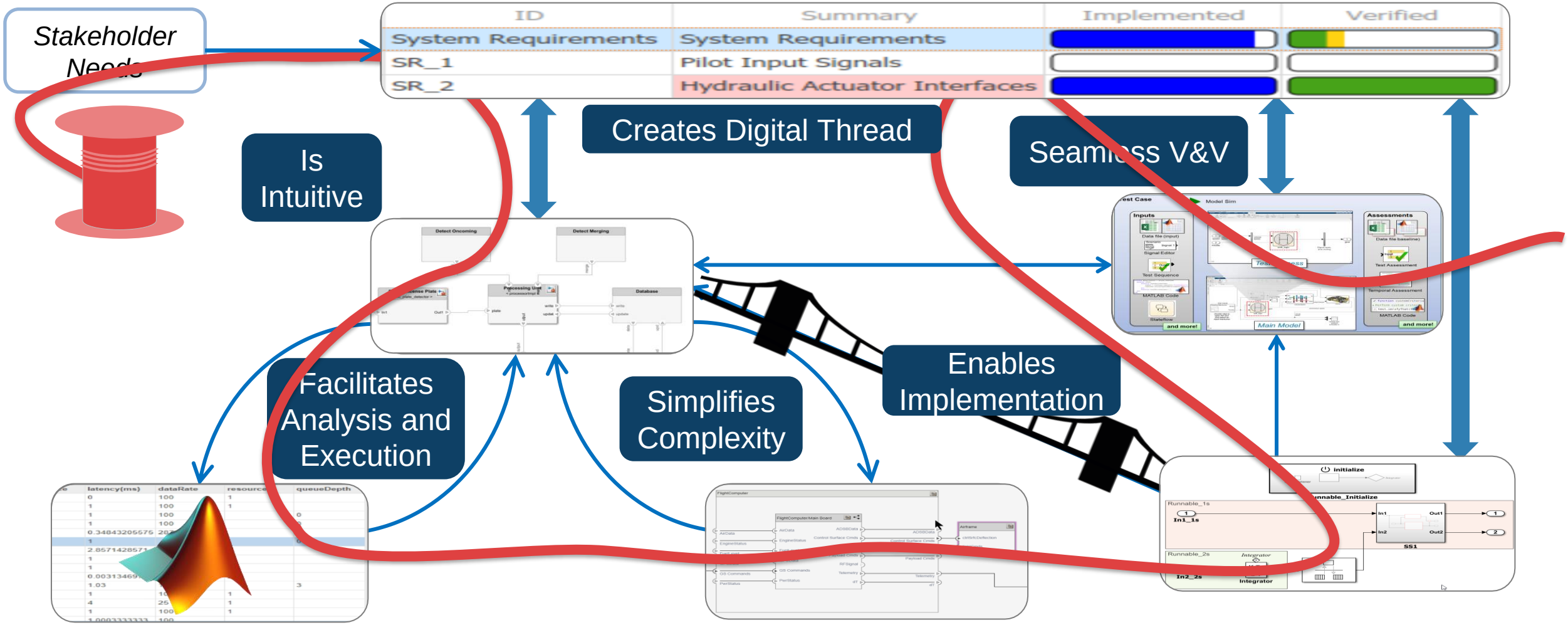
System Engineering/Development Workflow



System Engineering/Development Common Users Challenges



System Engineering/Development Unified Environment



System Engineering/Development Unified Environment

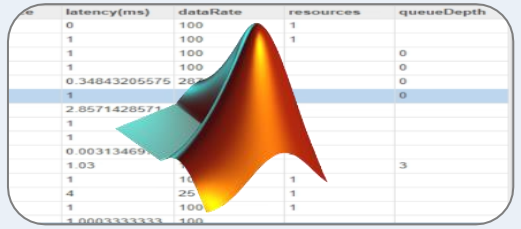
ID	Summary	Implemented	Verified
System Requirements	System Requirements		
SR_1	Pilot Input Signals		
SR_2	Hydraulic Actuator Interfaces		

Creates Digital Thread

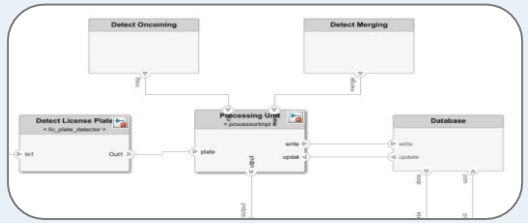
Requirements Toolbox

System Composer

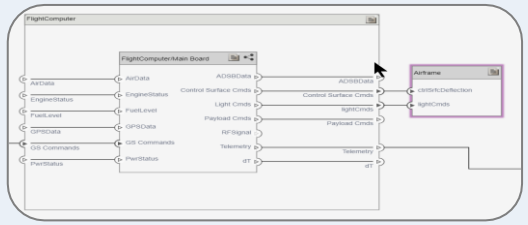
Facilitates Analysis and Execution



Is Intuitive



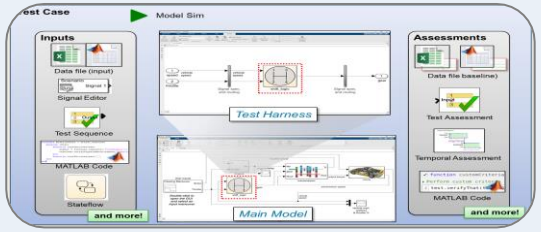
Simplifies Complexity



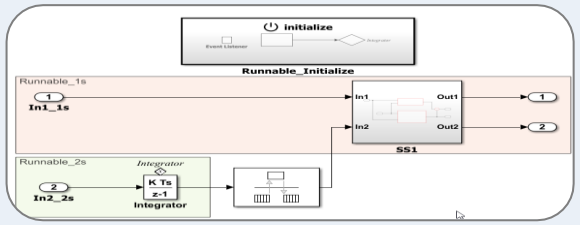
MATLAB

Simulink

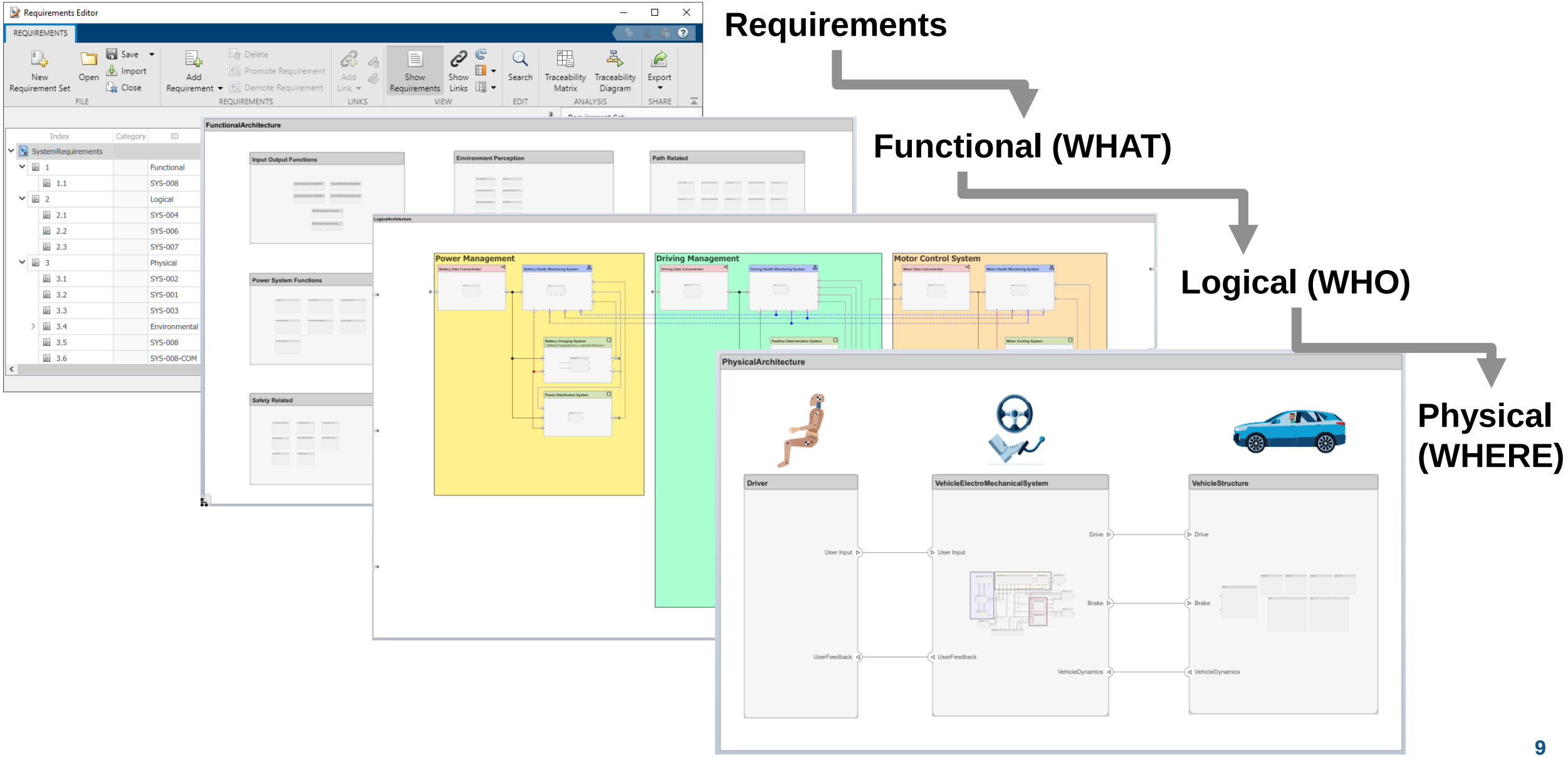
Seamless V&V



Enables Implementation

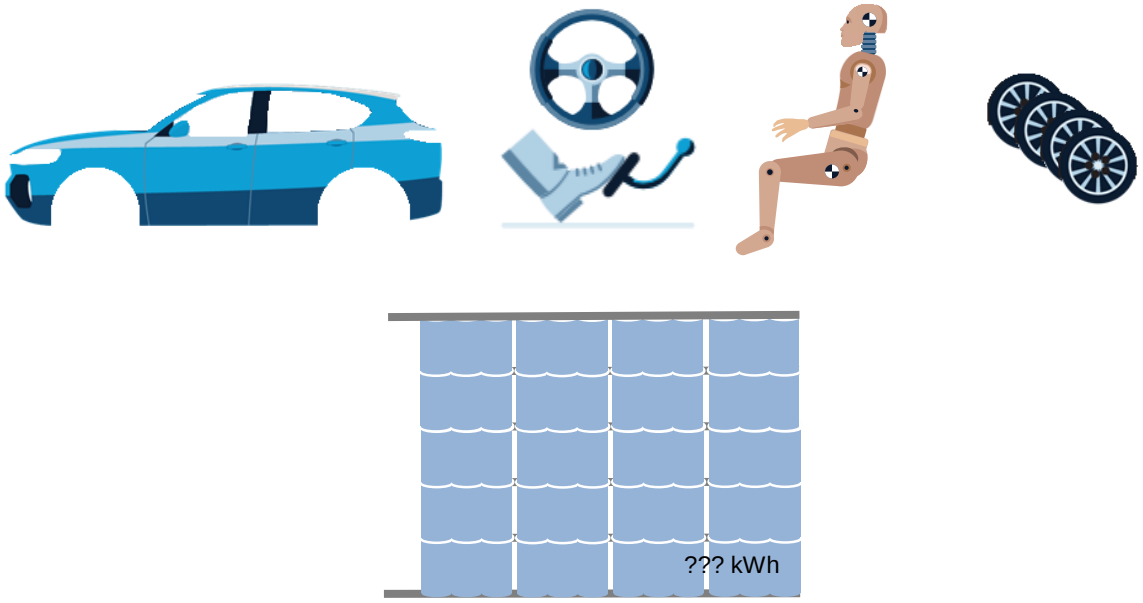


RFLP Architecture Modeling of an Electric Vehicle using System Composer



Analysis and Optimization of an EV Battery with Simulink Integration

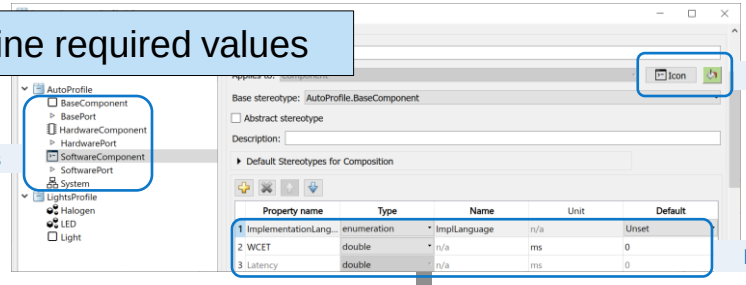
- Large Complex System
- Stakeholder Requirements
  - Cost
  - Weight
  - Range
- Known Power/Weight Budgets
- Unknown Battery Size



Stereotypes and Properties

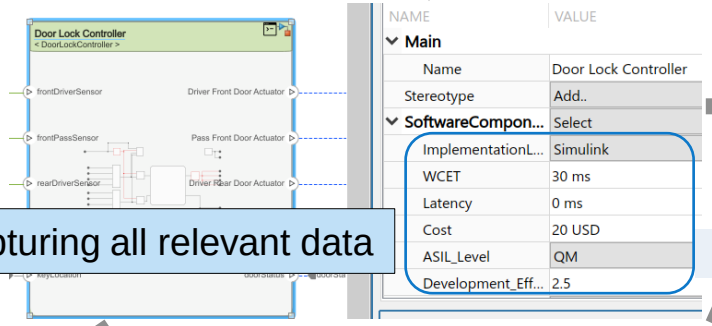
Define project-specific profiles with stereotypes, properties, and styling

Define required values



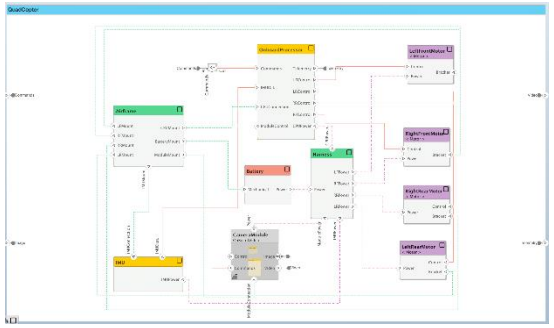
Select stereotypes and specify design properties for components, ports, connectors, and interfaces

Capturing all relevant data



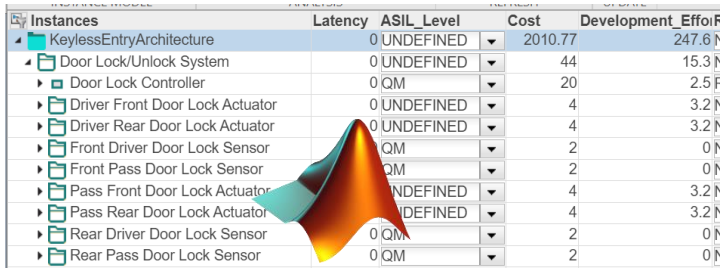
Improved readability

Semantic Styling (automatic)



Identifying best solution

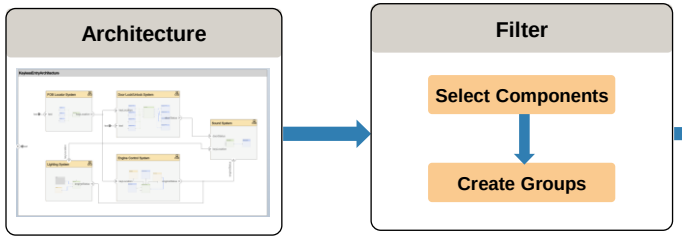
Perform analysis for trade off studies

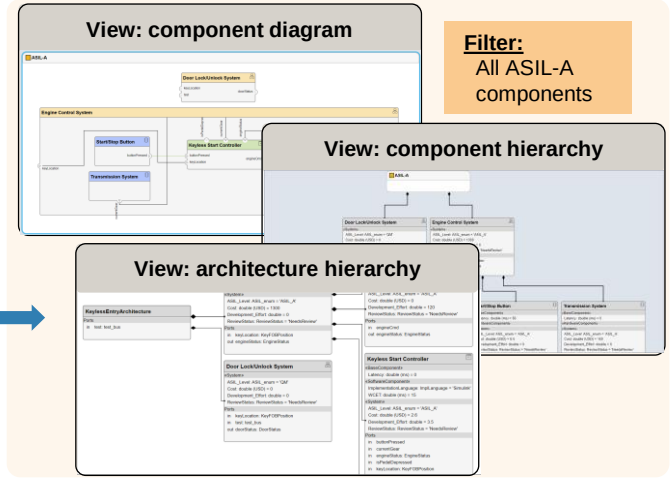


Instances	Latency	ASIL_Level	Cost	Development_Effor
KeylessEntryArchitecture	0	UNDEFINED	2010.77	247.6
Door Lock/Unlock System	0	UNDEFINED	44	15.3
Door Lock Controller	0	QM	20	2.5
Driver Front Door Lock Actuator	0	UNDEFINED	4	3.2
Driver Rear Door Lock Actuator	0	UNDEFINED	4	3.2
Front Driver Door Lock Sensor		QM	2	0
Front Pass Door Lock Sensor		QM	2	0
Pass Front Door Lock Actuator		UNDEFINED	4	3.2
Pass Rear Door Lock Actuator		UNDEFINED	4	3.2
Rear Driver Door Lock Sensor	0	QM	2	0
Rear Pass Door Lock Sensor	0	QM	2	0

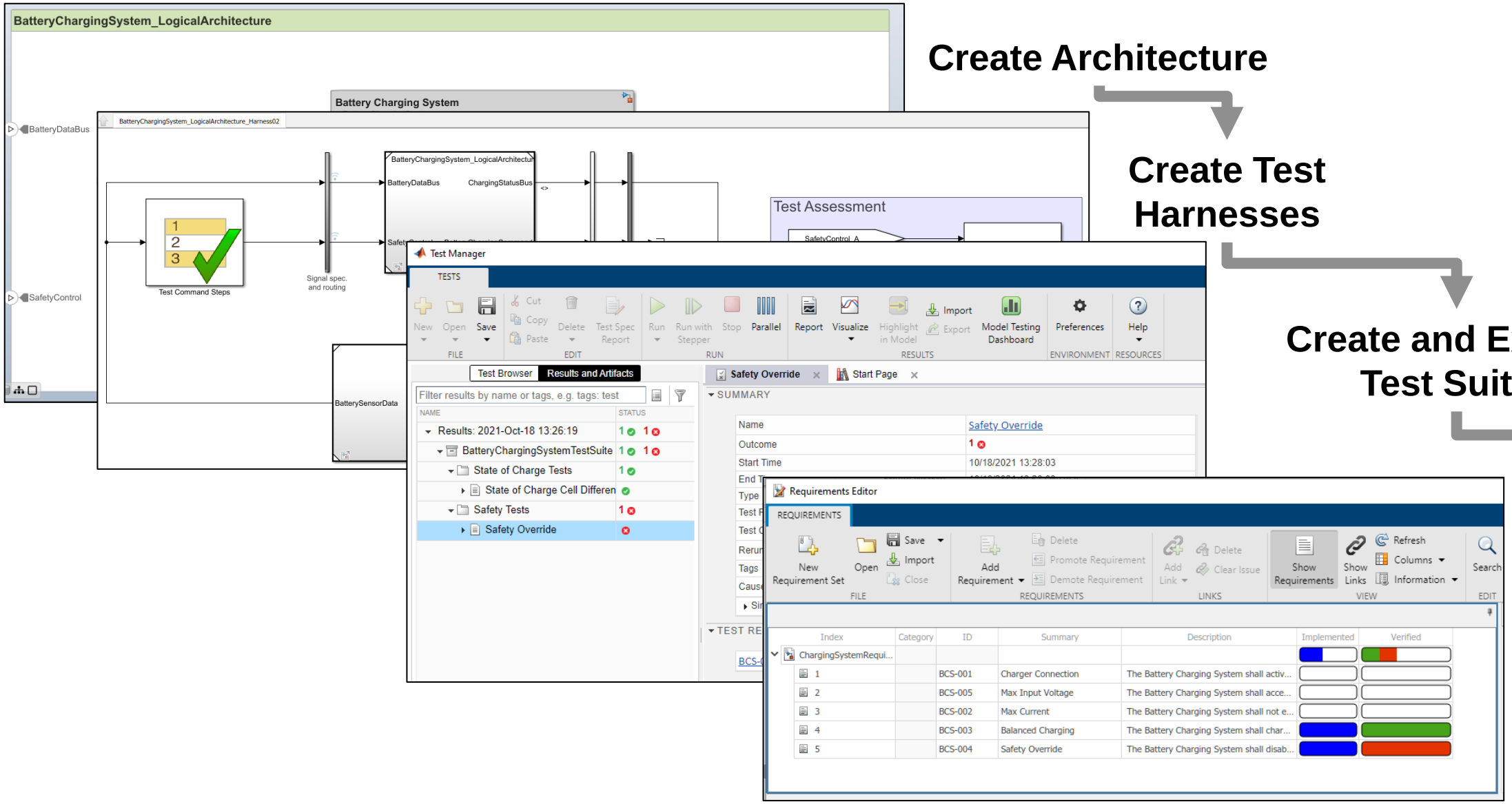
Focus on specific design concerns

Create automatic views to tackle complexity





Create Test Harnesses for System Requirements Verification



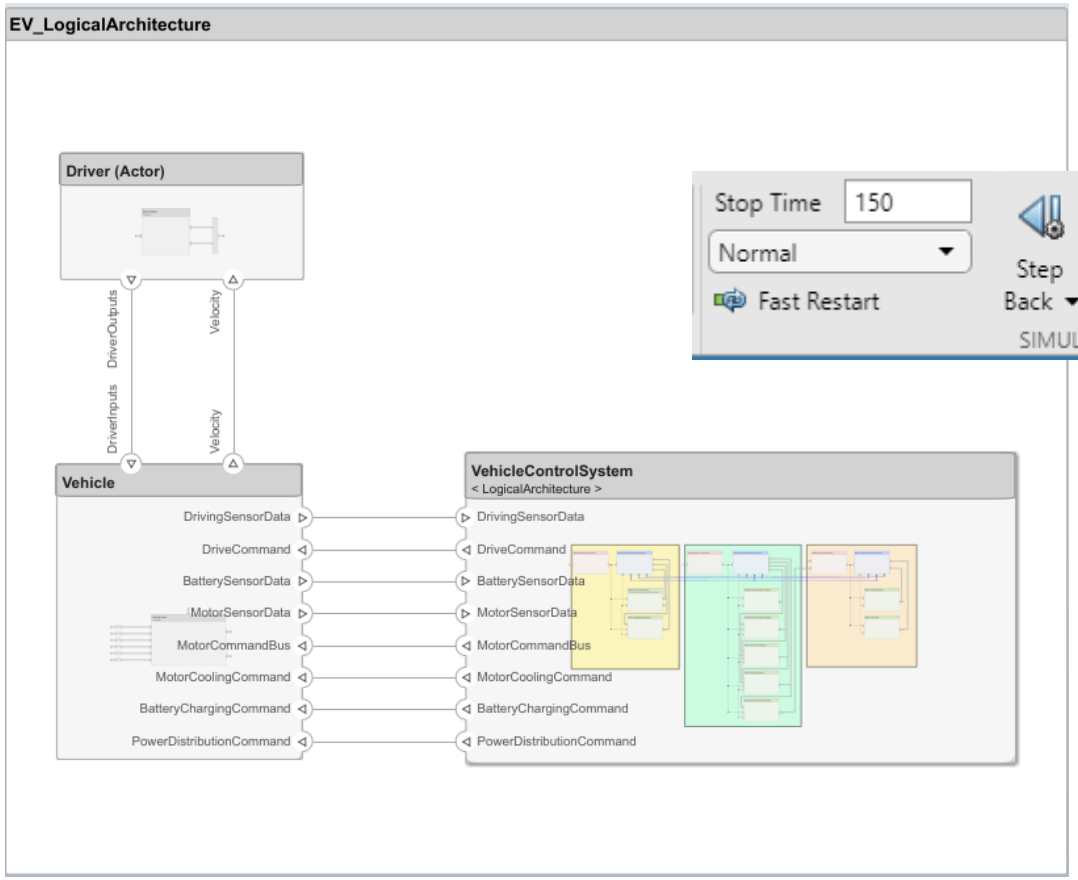
Create Architecture

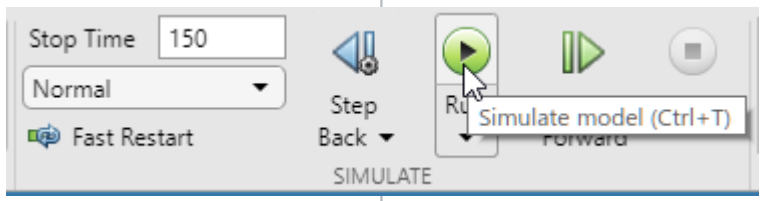
Create Test Harnesses

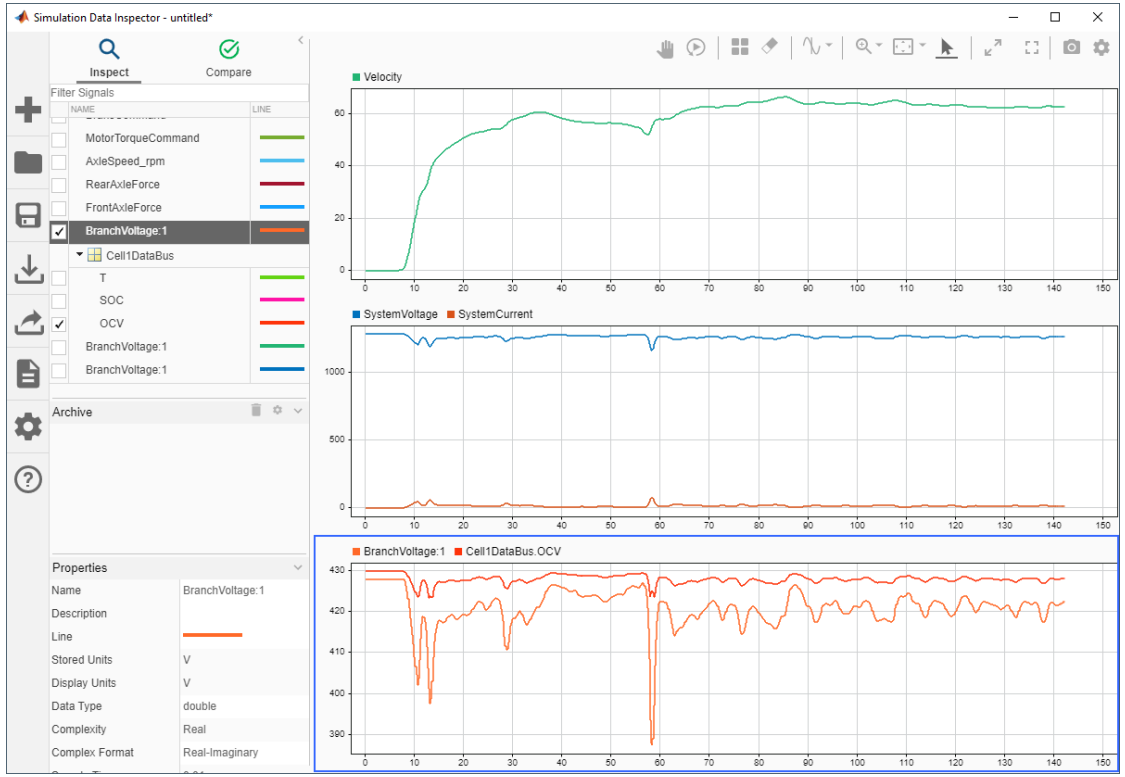
Create and Execute Test Suites

Assess Coverage

Simulate Integrated Architecture Models

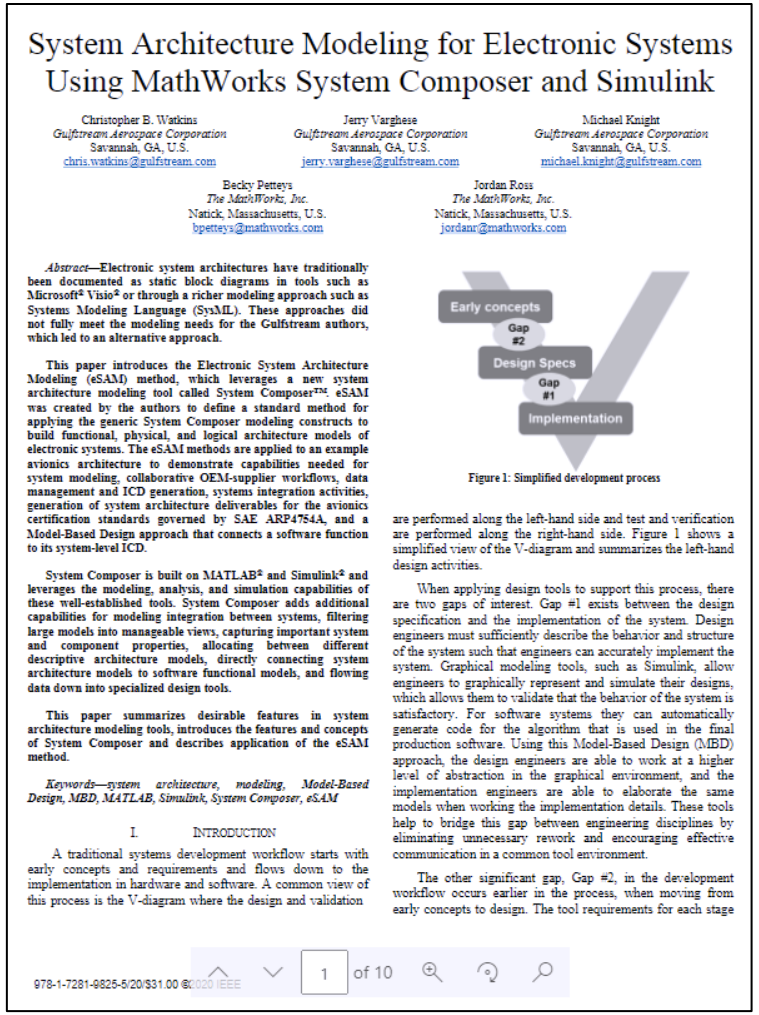






Gulfstream Using System Composer to Model Electronic System Architecture

"Goals of System Composer are to make system modeling easy, flexible, and scalable; and to ease the transition to the design environment. The features and constructs of System Composer reflect the prioritization of these goals."



System Architecture Modeling for Electronic Systems Using MathWorks System Composer and Simulink

Christopher B. Watkins
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
chris.watkins@gulfstream.com

Jerry Varghese
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
jerry.varghese@gulfstream.com

Michael Knight
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
michael.knight@gulfstream.com

Becky Petteys
The MathWorks, Inc.
Natick, Massachusetts, U.S.
bpetteys@mathworks.com

Jordan Ross
The MathWorks, Inc.
Natick, Massachusetts, U.S.
jordanr@mathworks.com

***Abstract*—Electronic system architectures have traditionally been documented as static block diagrams in tools such as Microsoft® Visio® or through a richer modeling approach such as Systems Modeling Language (SysML). These approaches did not fully meet the modeling needs for the Gulfstream authors, which led to an alternative approach.**

This paper introduces the Electronic System Architecture Modeling (eSAM) method, which leverages a new system architecture modeling tool called System Composer™. eSAM was created by the authors to define a standard method for applying the generic System Composer modeling constructs to build functional, physical, and logical architecture models of electronic systems. The eSAM methods are applied to an example avionics architecture to demonstrate capabilities needed for system modeling, collaborative OEM-supplier workflows, data management and ICD generation, systems integration activities, generation of system architecture deliverables for the avionics certification standards governed by SAE ARP4754A, and a Model-Based Design approach that connects a software function to its system-level ICD.

System Composer is built on MATLAB® and Simulink® and leverages the modeling, analysis, and simulation capabilities of these well-established tools. System Composer adds additional capabilities for modeling integration between systems, filtering large models into manageable views, capturing important system and component properties, allocating between different descriptive architecture models, directly connecting system architecture models to software functional models, and flowing data down into specialized design tools.

This paper summarizes desirable features in system architecture modeling tools, introduces the features and concepts of System Composer and describes application of the eSAM method.

***Keywords*—system architecture, modeling, Model-Based Design, MBD, MATLAB, Simulink, System Composer, eSAM**

I. INTRODUCTION

A traditional systems development workflow starts with early concepts and requirements and flows down to the implementation in hardware and software. A common view of this process is the V-diagram where the design and validation

Figure 1: Simplified development process

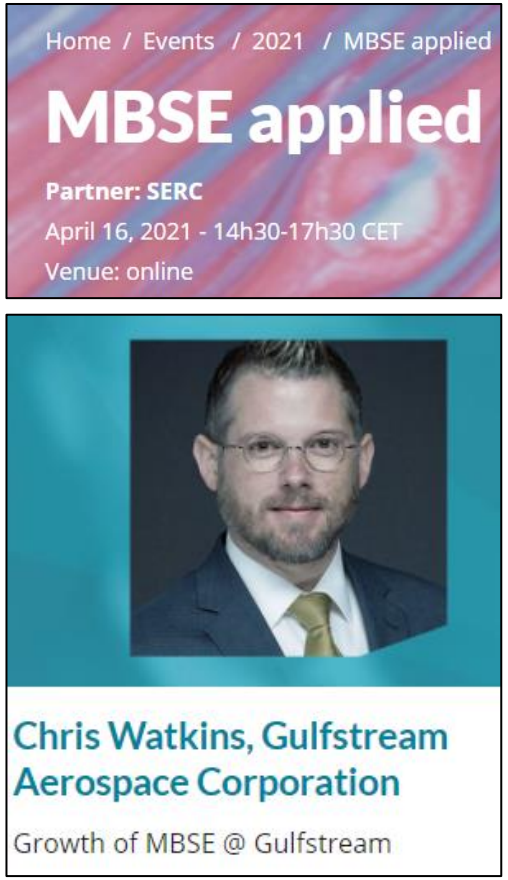
are performed along the left-hand side and test and verification are performed along the right-hand side. Figure 1 shows a simplified view of the V-diagram and summarizes the left-hand design activities.

When applying design tools to support this process, there are two gaps of interest. Gap #1 exists between the design specification and the implementation of the system. Design engineers must sufficiently describe the behavior and structure of the system such that engineers can accurately implement the system. Graphical modeling tools, such as Simulink, allow engineers to graphically represent and simulate their designs, which allows them to validate that the behavior of the system is satisfactory. For software systems they can automatically generate code for the algorithm that is used in the final production software. Using this Model-Based Design (MBD) approach, the design engineers are able to work at a higher level of abstraction in the graphical environment, and the implementation engineers are able to elaborate the same models when working the implementation details. These tools help to bridge this gap between engineering disciplines by eliminating unnecessary rework and encouraging effective communication in a common tool environment.

The other significant gap, Gap #2, in the development workflow occurs earlier in the process, when moving from early concepts to design. The tool requirements for each stage

978-1-7281-9825-5/20/$31.00 ©2020 IEEE

Link to the Paper



PDF
Video

DENSO Builds System Architecture Model for Auxiliary Motor to Accelerate Control Design and Verification

Challenge

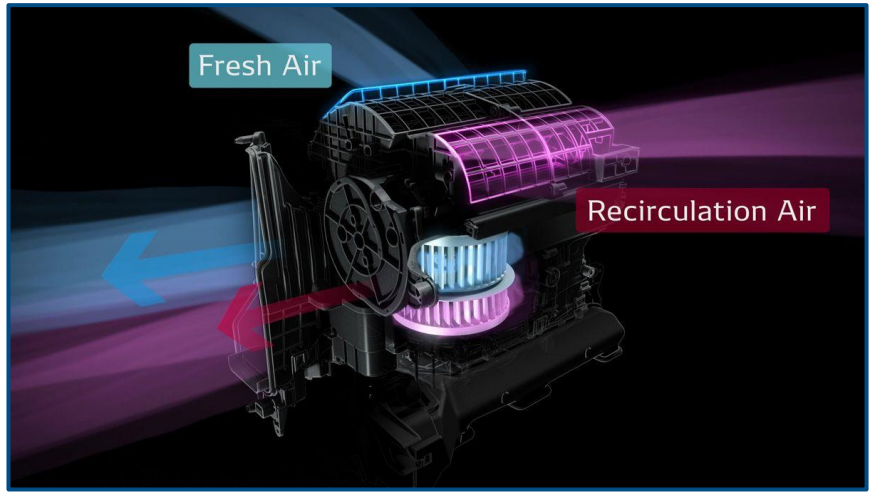
Model and analyze the core and customized parts of auxiliary motors separately before they are integrated

Solution

Use System Composer as the system architecture and use Simulink to model the customized component of the auxiliary motor

Results

- Workload reduced by one-third while maintaining high quality
- Model-Based Design process expanded to multiple products
- Automotive SPICE-like development process enabled with a single tool



DENSO blower motors deliver hot and cool air through a vehicle HVAC system.

"With Simulink and System Composer, we were able to efficiently create a design environment with a higher level of abstraction for the model-based systems engineering domain."

- Kazuyuki Hirai, DENSO Corporation

Link to user story

Conclusion

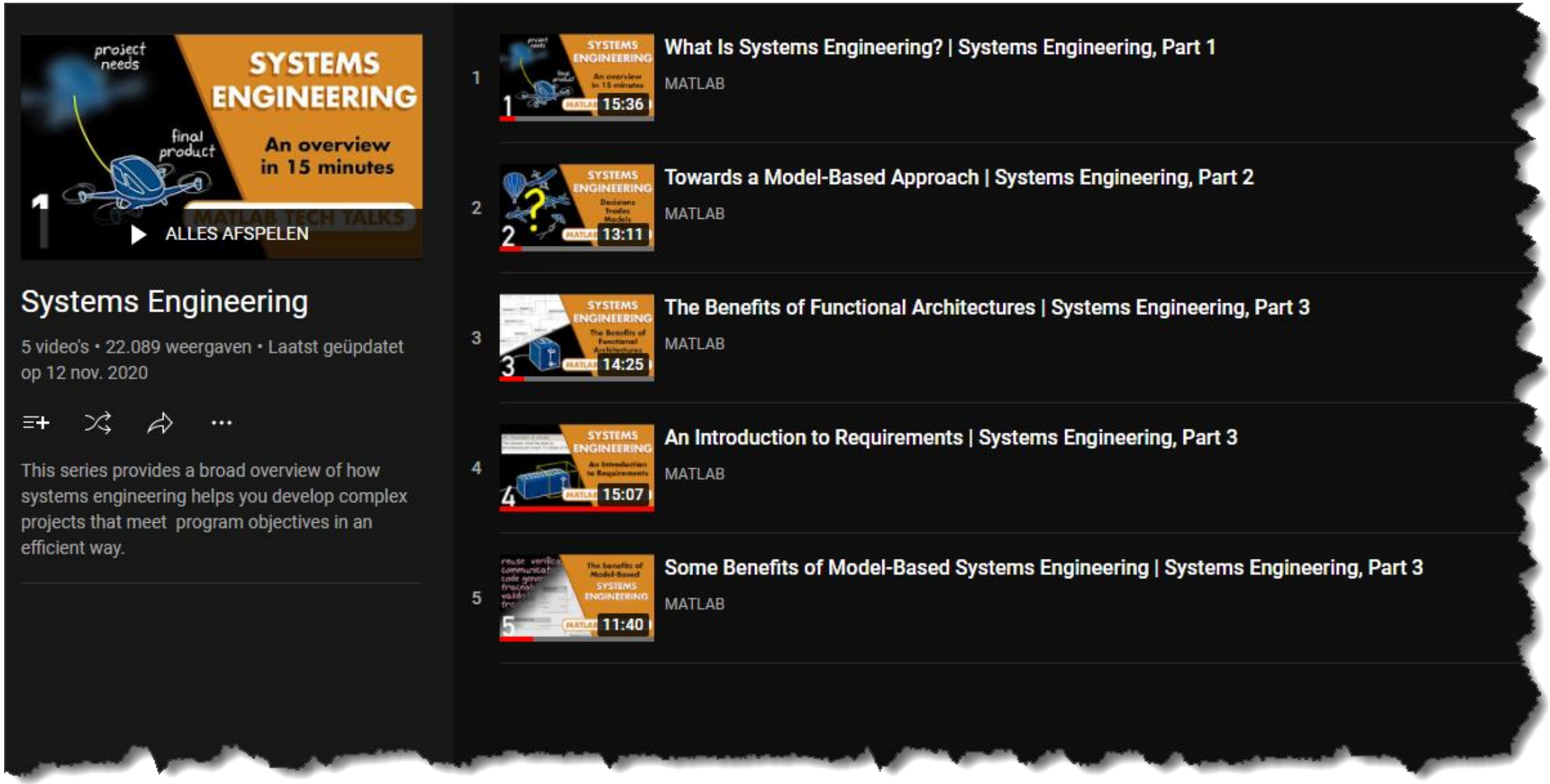
1. Capture and manage stakeholder's **needs** and describe system architectures
2. **Linking** requirements, system architectures, simulation models, and perform **trade-off analysis**
3. Achieve verification and validation through **simulation**

MBSE with System Composer in use today

Learn how these tools are being used today

- 2023 Denso - Architecture Model to Accelerate Control Design and Verification
- 2022 Gulfstream - MATLAB Expo keynote
- 2022 Bosch, India – Safety analysis for Product Development
- 2022 Tata Consultancy Services, India
- 2022 Ford, US: Building a digital thread from MBD to MBSE for ISO 26262
- 2021 Bosch, Germany: Architecture Design according to Automotive SPICE
- 2021 Mercedes Benz, India: Architecture Creation

Systems Engineering introduction by Brian Douglas



https://www.youtube.com/playlist?list=PLn8PRpmsu08owzDpgnQr7vo2O-FUQm_fL

System Engineering: From Requirements to Architecture to Simulation

System Engineering: From Requirements to Architecture to Simulation

Engineers use model-based systems engineering (MBSE) to manage system complexity, improve communication, and produce optimized system performance. Successful MBSE requires the synthesis of stakeholder requirements into architecture models to create intuitive system descriptions.

MATLAB®, Simulink®, System Composer™, and Simulink Requirements™ together provide a platform to create descriptive architecture models that seamlessly bridge into detailed implementation models. The connected environment ensures items across the architecture and design worlds stay in sync. Systems engineers can establish a digital thread to navigate between system requirements, architecture models, implementation models, and embedded software.

Access these resources and learn how you can:

- ✔ Capture and manage system requirements enabling impact and coverage analysis
- ✔ Optimize system architectures by capturing architecture metadata and directly connecting to MATLAB analytics for domain-specific trade studies
- ✔ Create simplified model views to isolate the components of interest for different engineering concerns
- ✔ Validate requirements and verify system architectures using simulation-based tests
- ✔ Translate and refine requirements into architectures with components ready for simulation and implementation using Model-Based Design in Simulink

30-Day Free Trial

Try MATLAB, Simulink, and more.

» Get started

https://www.mathworks.com/campaigns/offers/model-based-system-engineering.html

MATLAB EXPO

Thank you

MathWorks®