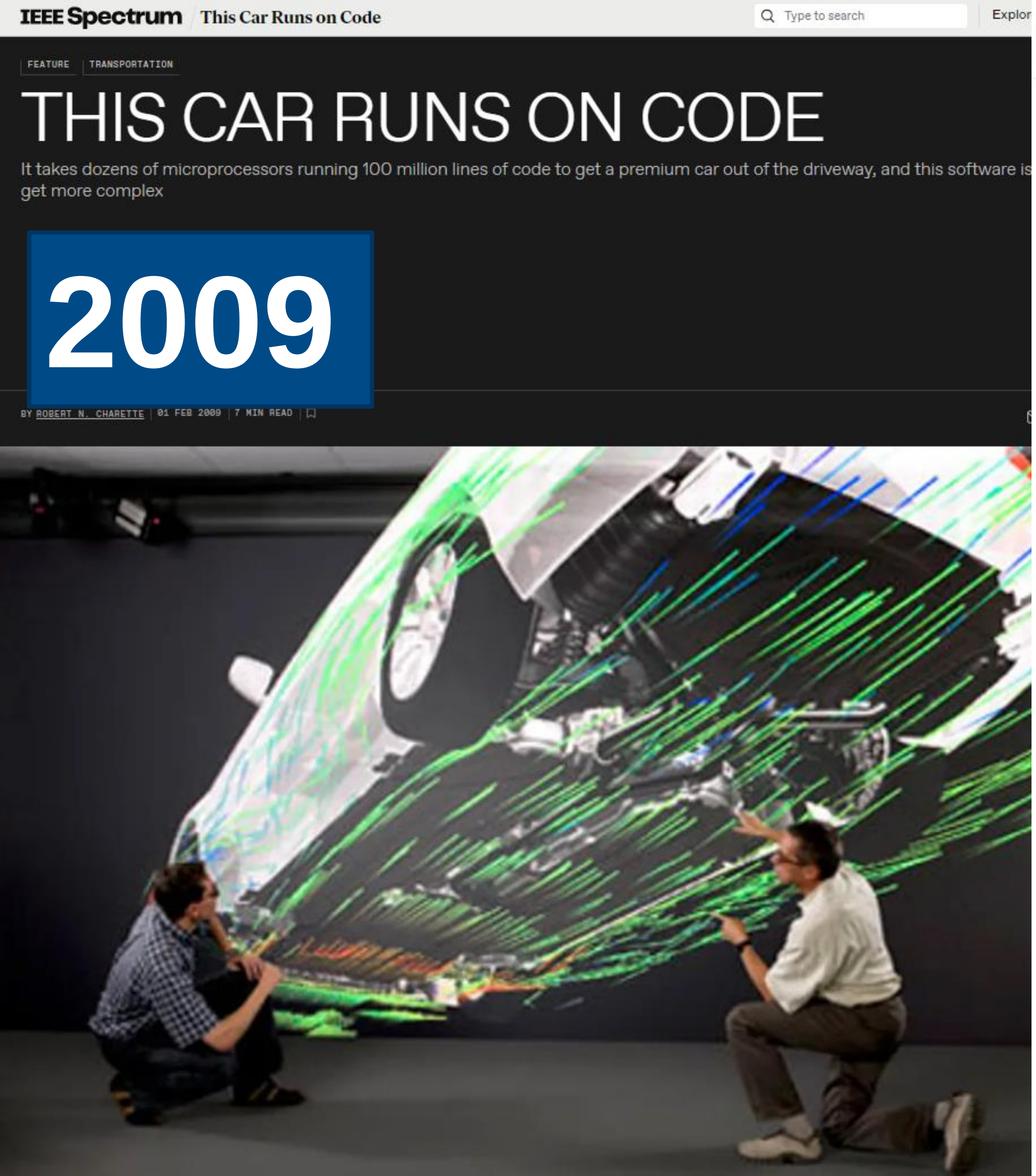
MathWorks
AUTOMOTIVE CONFERENCE 2023
Europe

Software Architectures and Virtual Integration in Model-Based Design

Mani Ramamurthy, MathWorks

MathWorks



IEEE Spectrum | This Car Runs on Code

Type to search | Explor

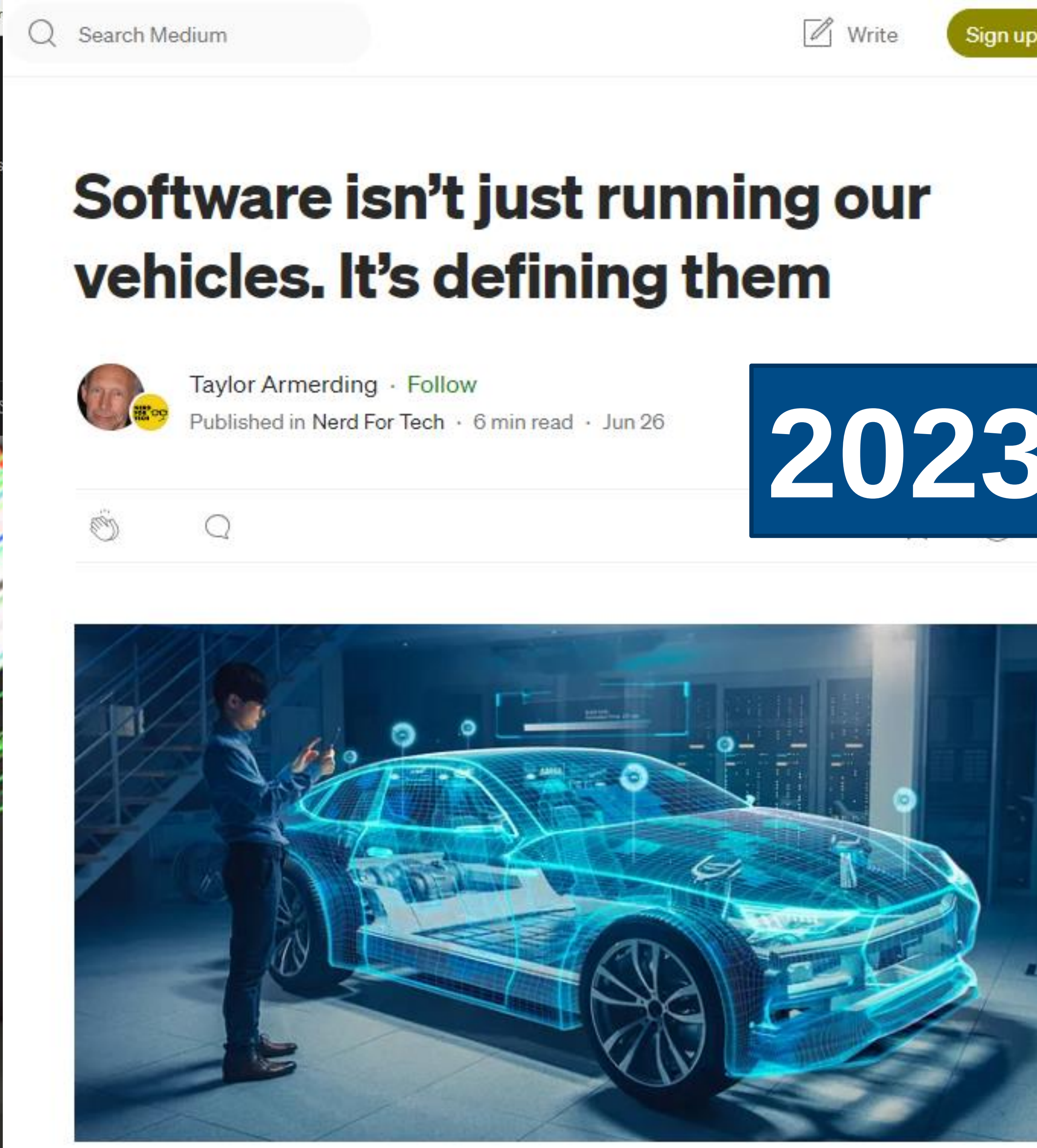
FEATURE | TRANSPORTATION

THIS CAR RUNS ON CODE

It takes dozens of microprocessors running 100 million lines of code to get a premium car out of the driveway, and this software is
get more complex

BY ROBERT N. CHARETTE | 01 FEB 2009 | 7 MIN READ

2009



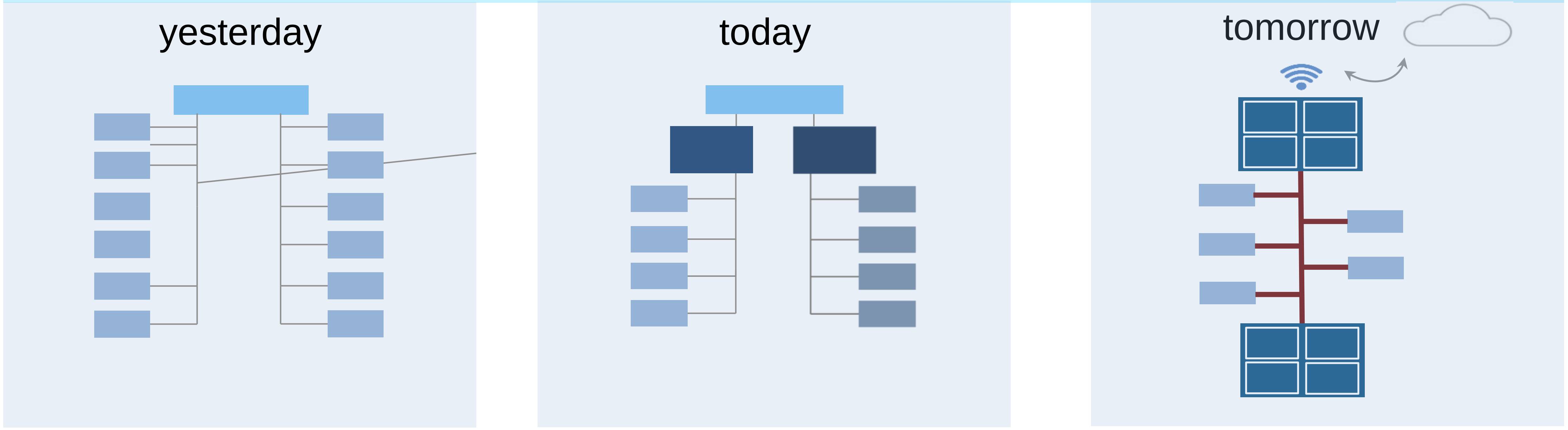
Search Medium | Write | Sign up

Software isn't just running our vehicles. It's defining them

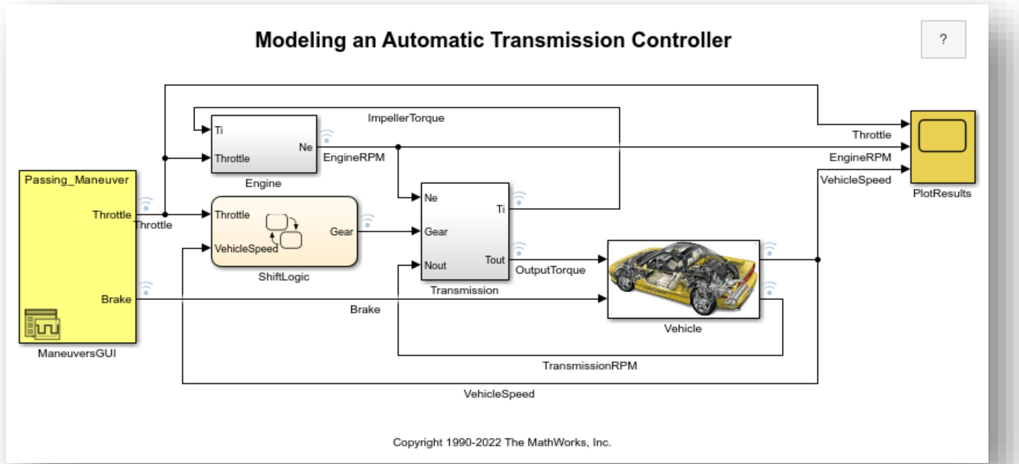
Taylor Armerding · Follow
Published in Nerd For Tech · 6 min read · Jun 26

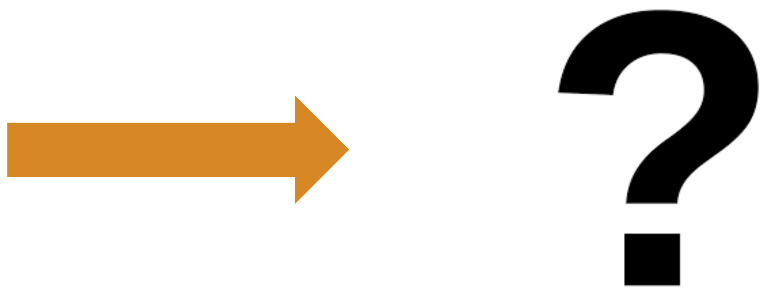
2023

E/E architectures



Modeling & Design Tools

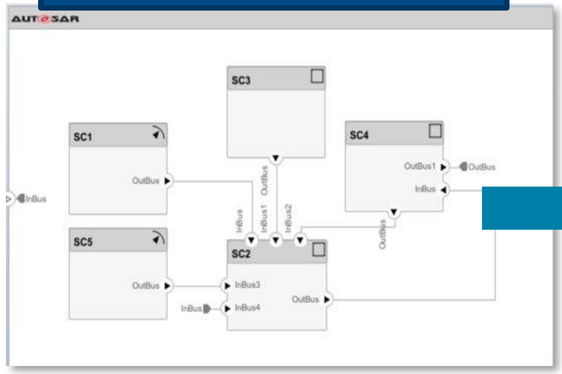




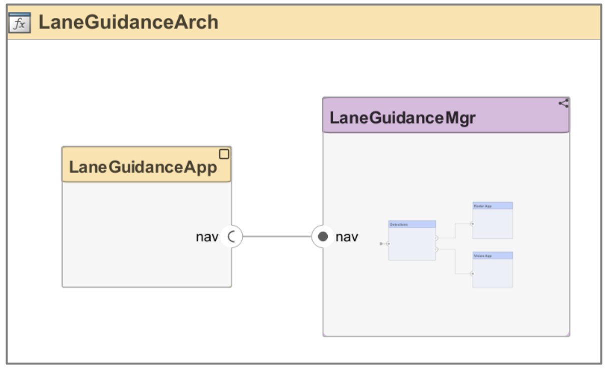
How is the Simulink Platform staying relevant?

- Models → Architectures

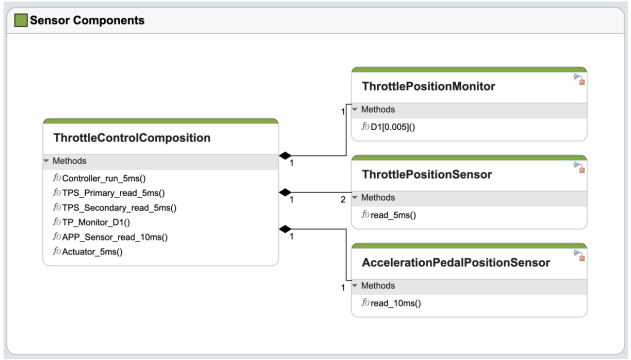
ARCHITECTURE



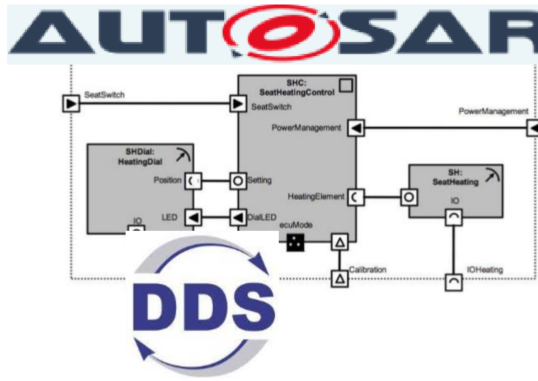
SOA



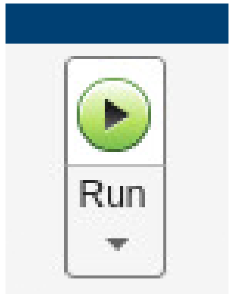
OO PRINCIPLES



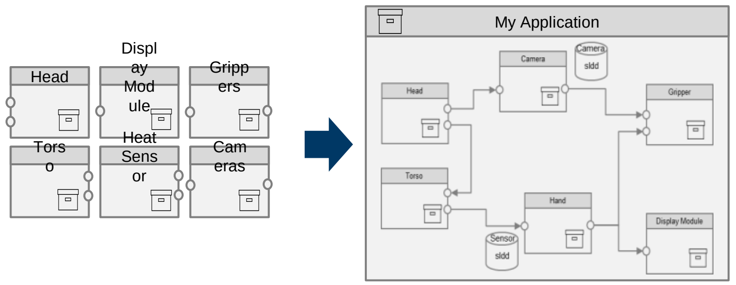
FRAMEWORKS



- Simulation → Virtual Integration



AUTO ASSEMBLY



CODE CENTRIC DESIGN

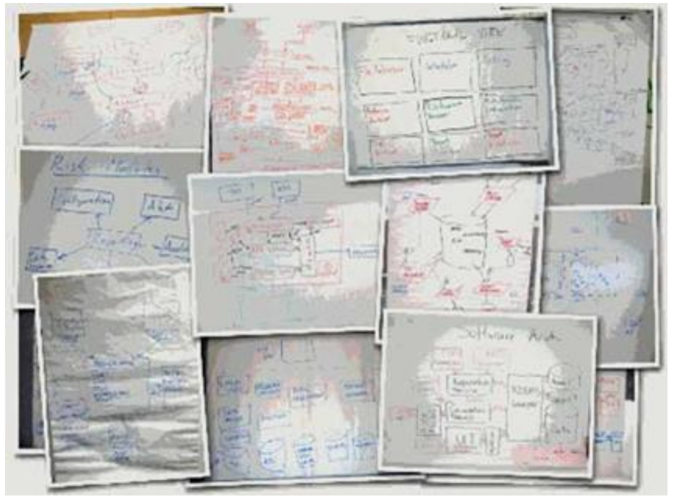
```
class CodeComponent final
{
 public:

  // service function f0
  void f0(real_T rtu_u, real_T
*rty_y);

  // service function f1
```

CI

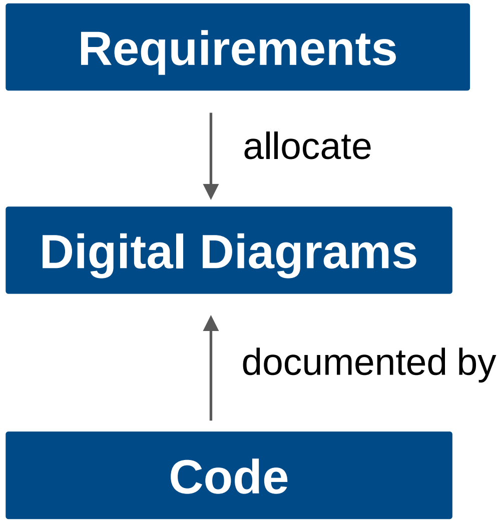
From Models to Architectures

Software architectures are abstractions to get good implementations

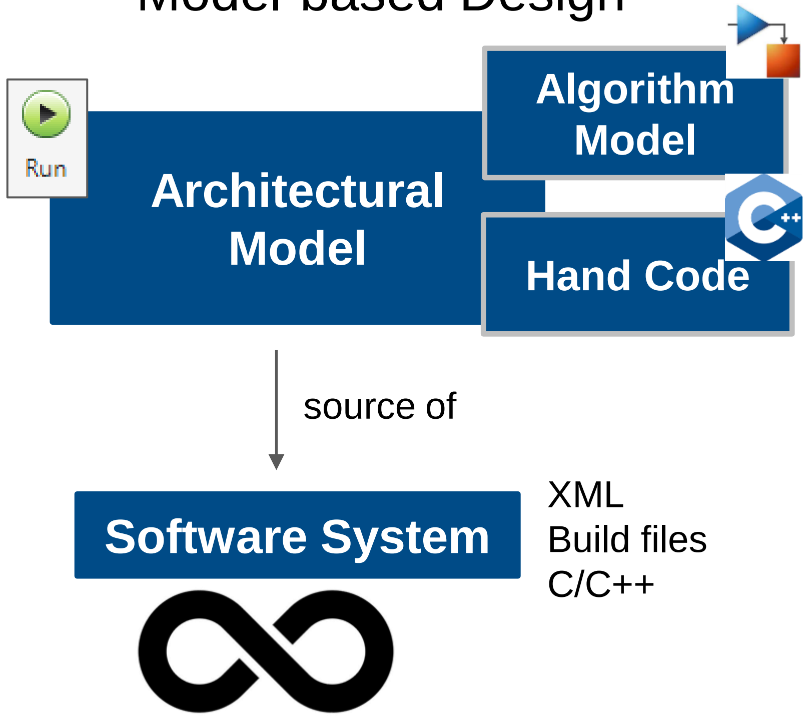
Facilitate the creative design process



Conform to digital engineering



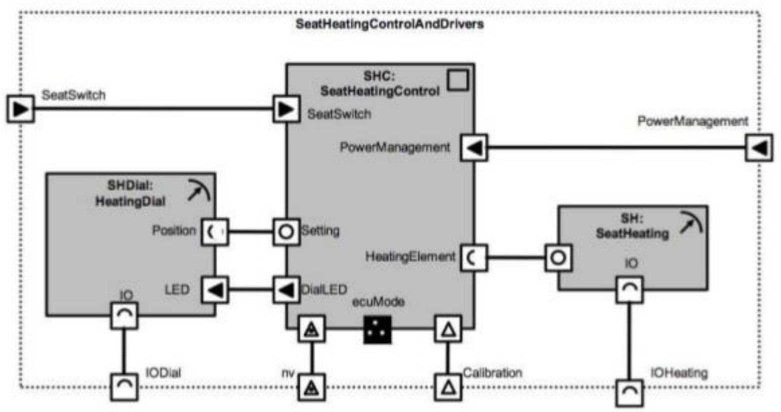
Implement leveraging Model-based Design



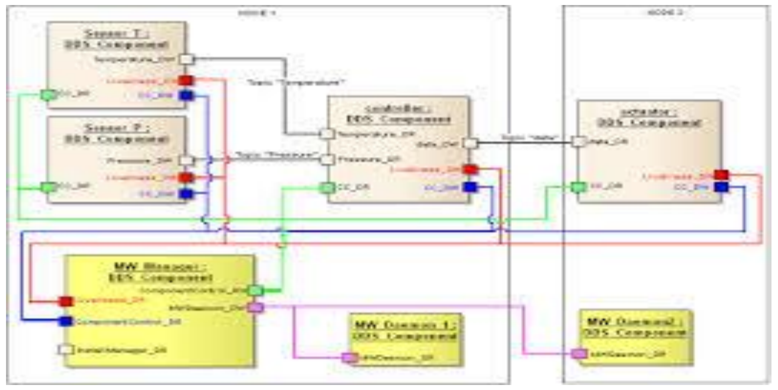
Low barrier of entry

Completes deep workflow

Component-Port-Connector Diagrams are standard representations of Software Architectures

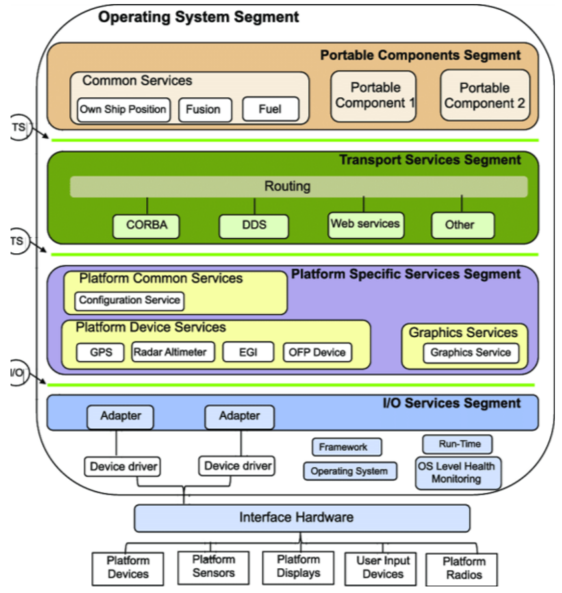


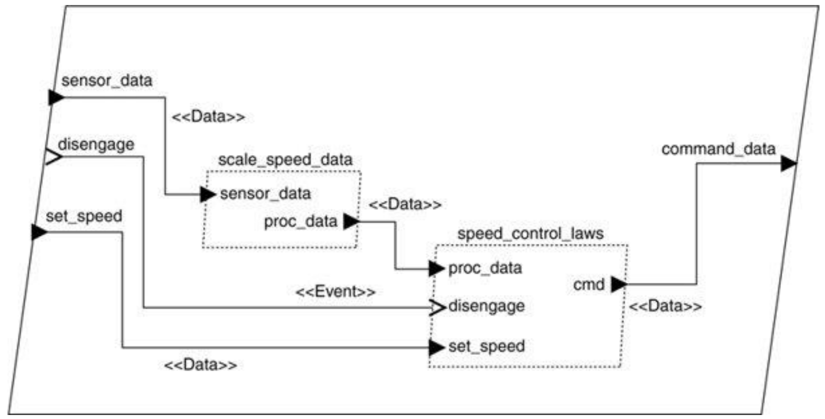
AUTOSAR



DDS

Future Airborne Capability Environment

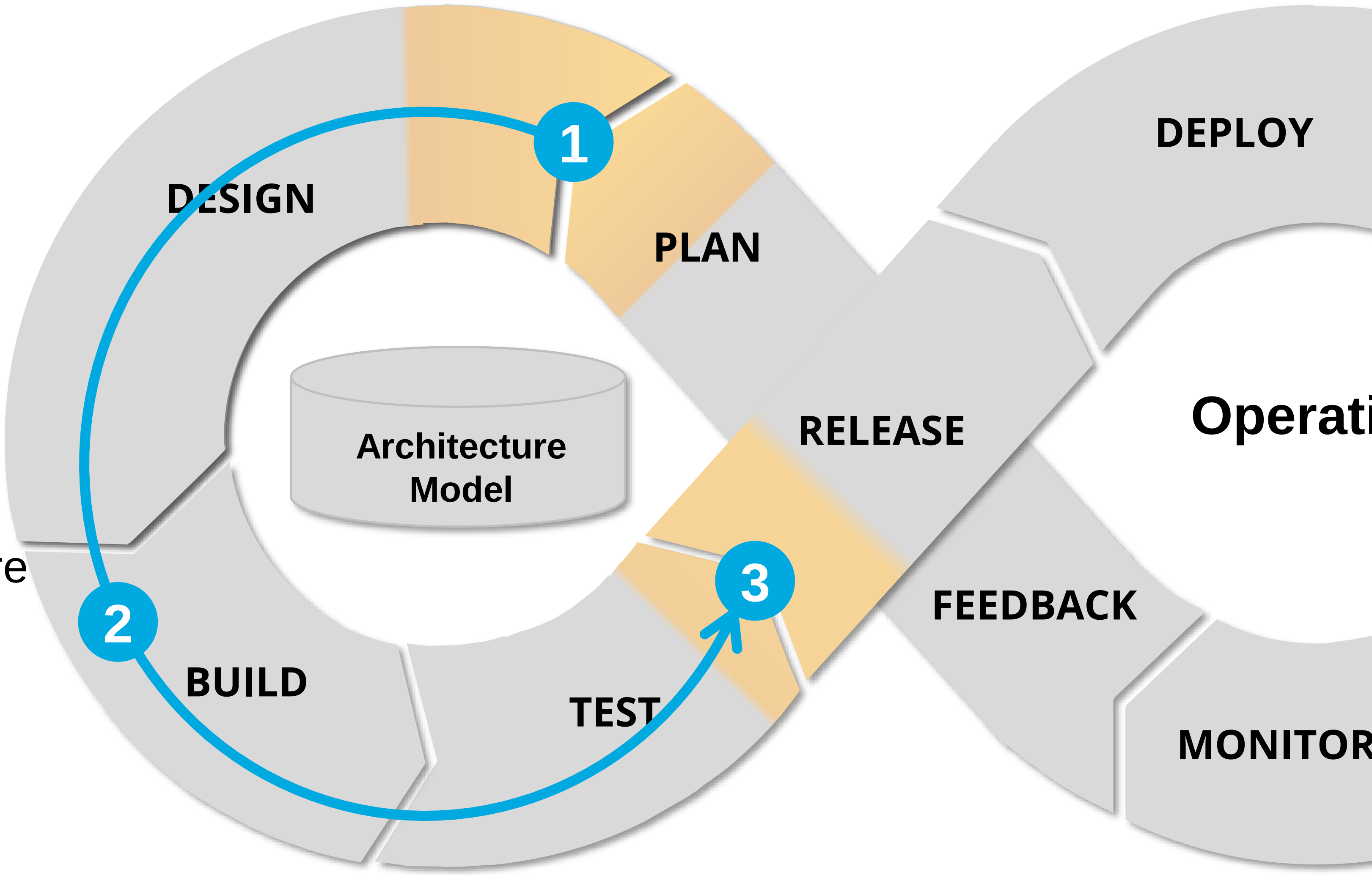


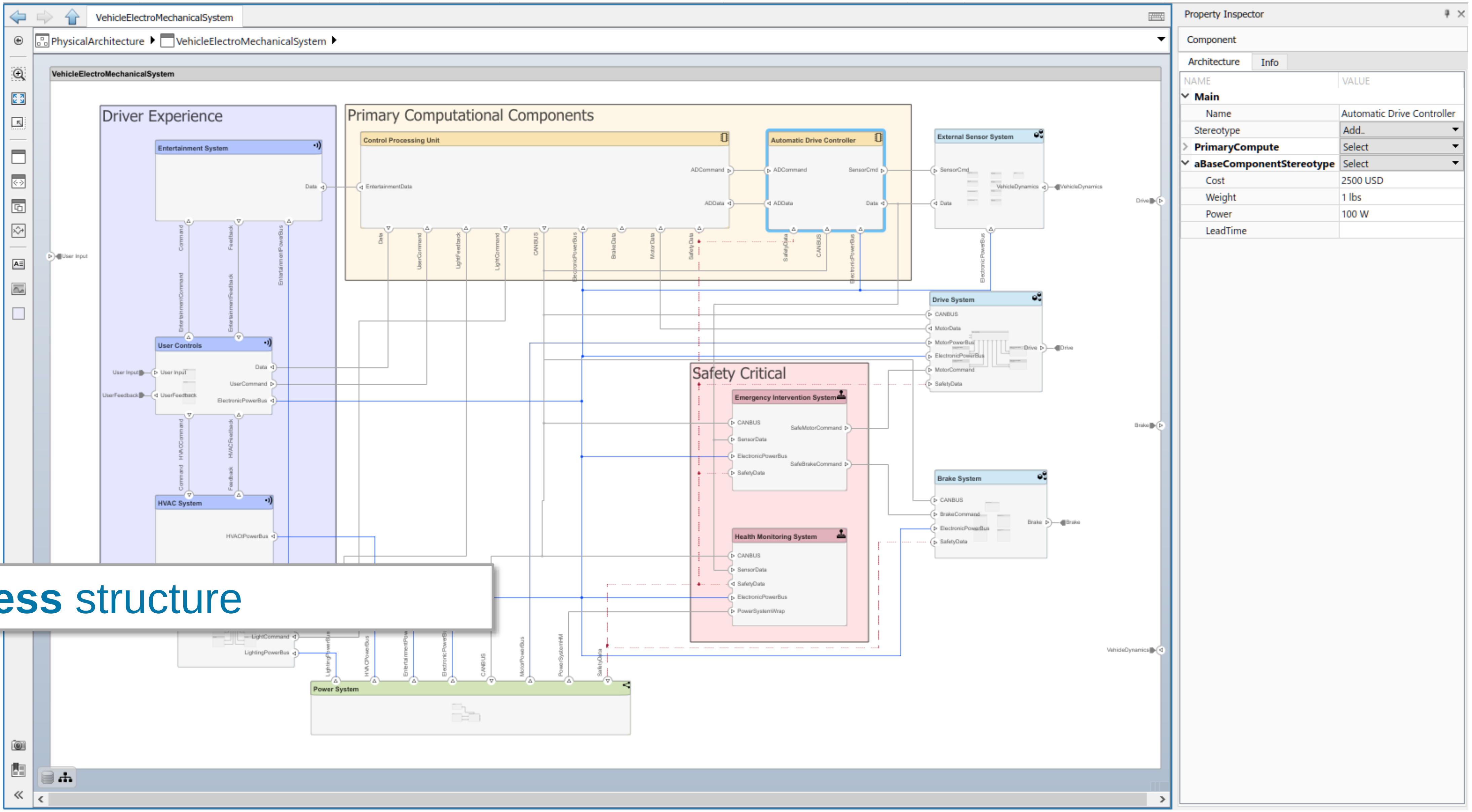


AADL

System Composer is our platform for Architecture Modeling

1. Expressive & intuitive architecture modeling language
2. Seamless and Collaborative workflow between MBSE, Software Architecture and MBD
3. Rigor for virtual system integration, test, and implementation

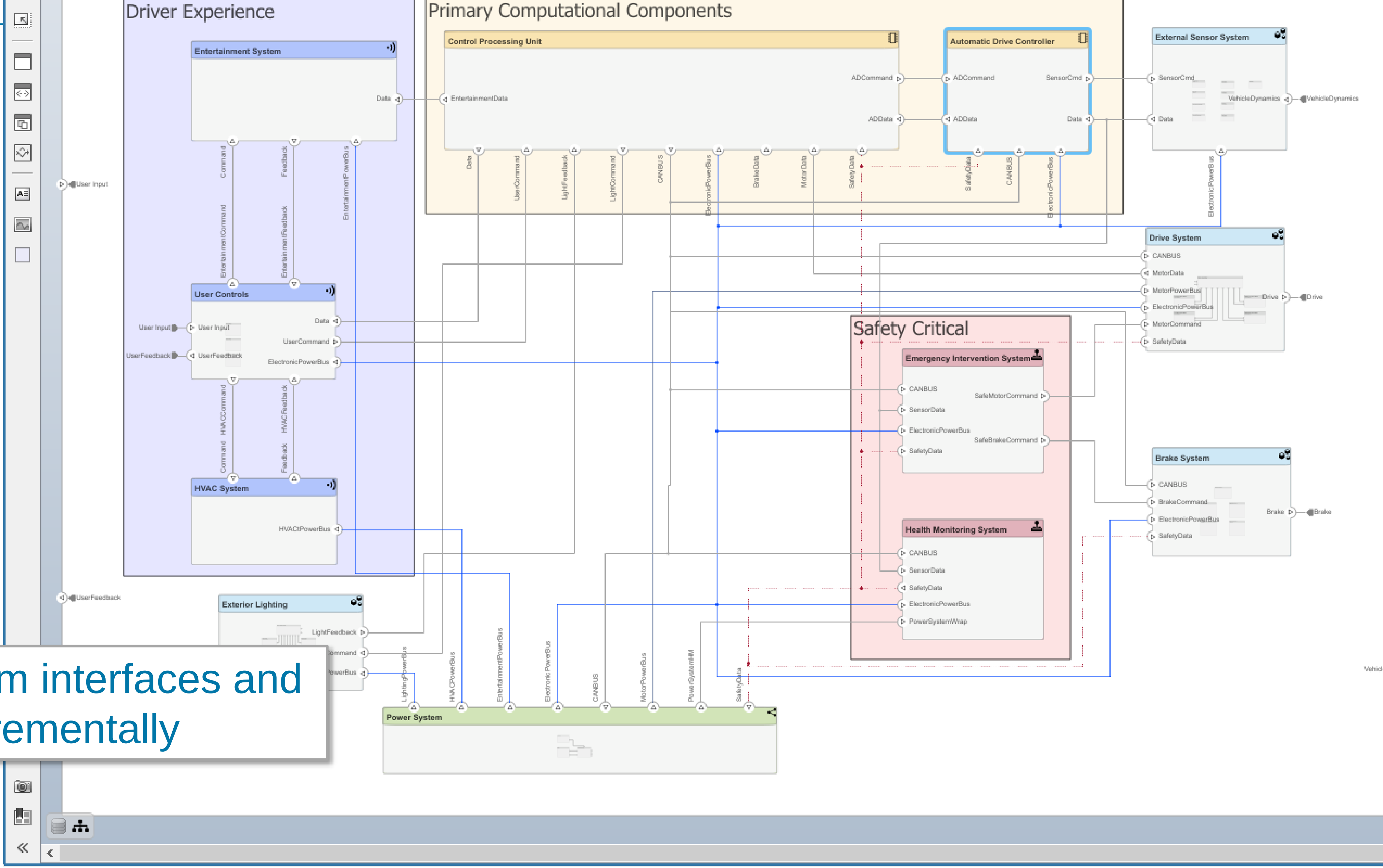




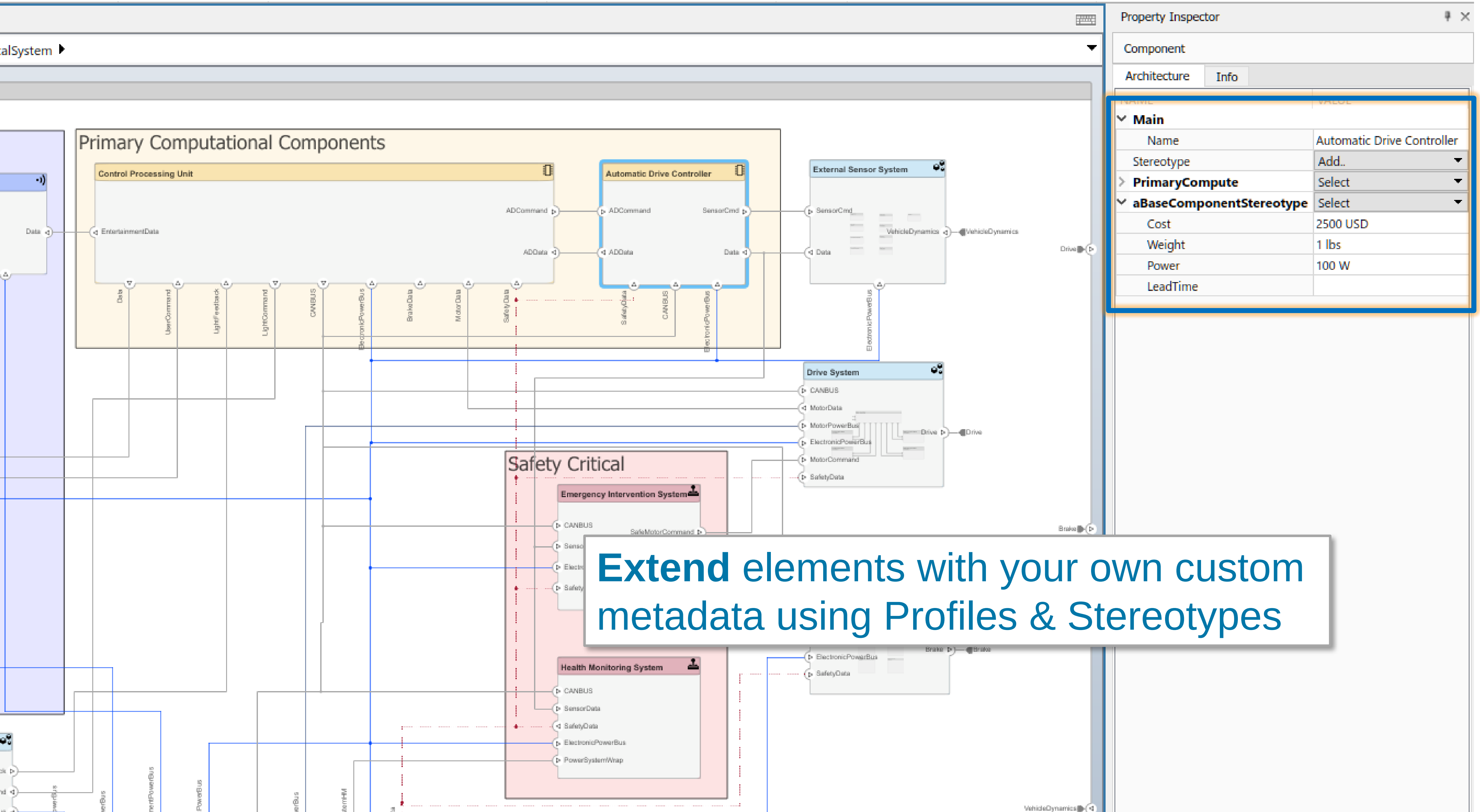
Property Inspector

NAME	VALUE
Main	
Name	Automatic Drive Controller
Stereotype	Add..
PrimaryCompute	Select
aBaseComponentStereotype	Select
Cost	2500 USD
Weight	1 lbs
Power	100 W
LeadTime	

Express structure



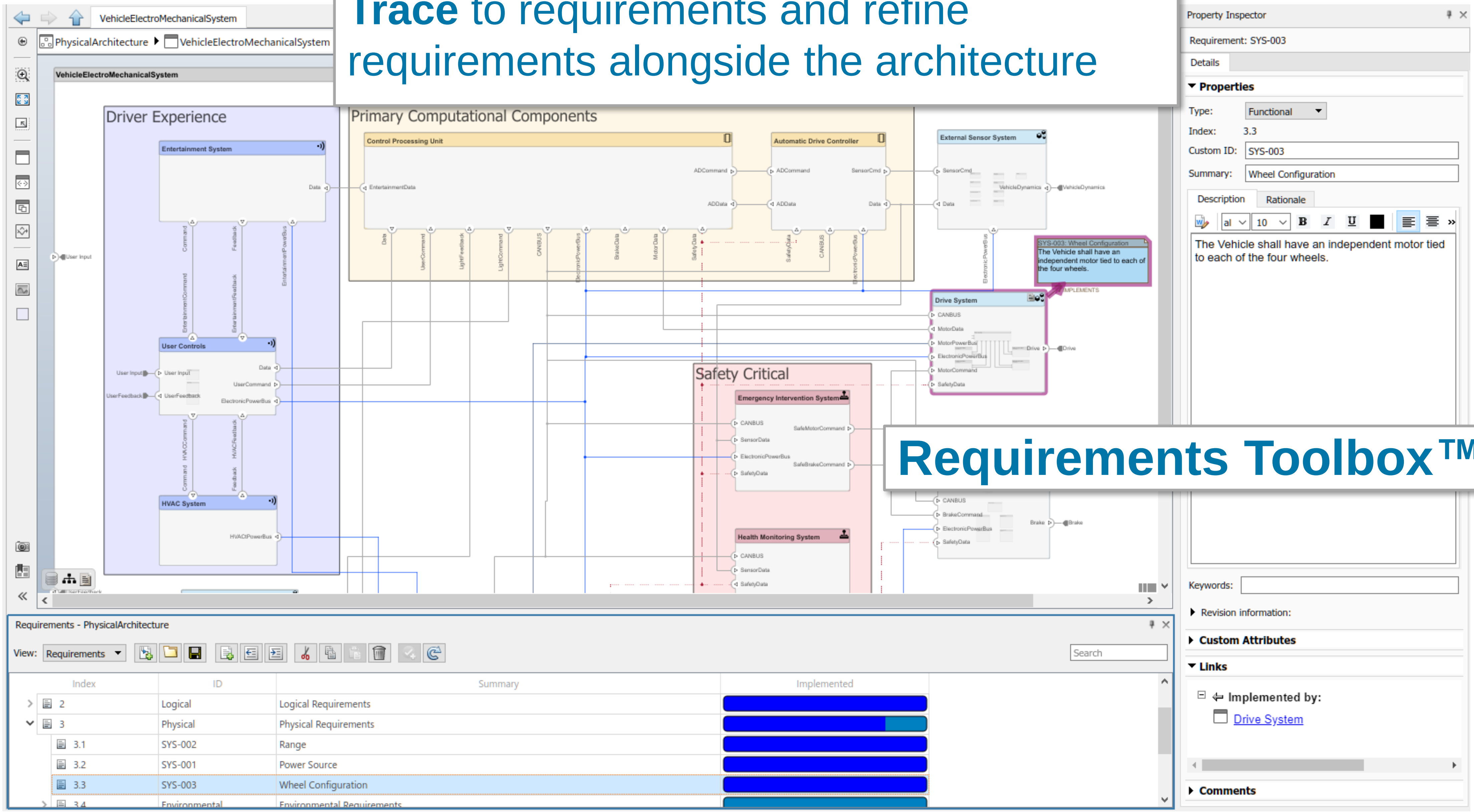
Sketch system interfaces and elaborate incrementally



Main	
Name	Automatic Drive Controller
Stereotype	Add..
PrimaryCompute	Select
aBaseComponentStereotype	Select
Cost	2500 USD
Weight	1 lbs
Power	100 W
LeadTime	

Extend elements with your own custom metadata using Profiles & Stereotypes

Trace to requirements and refine requirements alongside the architecture


VehicleElectroMechanicalSystem
PhysicalArchitecture
Driver Experience
Entertainment System
User Controls
HVAC System
Primary Computational Components
Control Processing Unit
Automatic Drive Controller
External Sensor System
SYS-003: Wheel Configuration
The Vehicle shall have an independent motor tied to each of the four wheels.
IMPLEMENTS
Drive System
Safety Critical
Emergency Intervention System
Health Monitoring System

Requirements - PhysicalArchitecture

View: Requirements

Index	ID	Summary	Implemented
2	Logical	Logical Requirements	
3	Physical	Physical Requirements	
3.1	SYS-002	Range	
3.2	SYS-001	Power Source	
3.3	SYS-003	Wheel Configuration	
3.4	Environmental	Environmental Requirements	

Property Inspector

Requirement: SYS-003

Details

Properties

Type: Functional

Index: 3.3

Custom ID: SYS-003

Summary: Wheel Configuration

Description | Rationale

The Vehicle shall have an independent motor tied to each of the four wheels.

Keywords:

Revision information:

Custom Attributes

Links

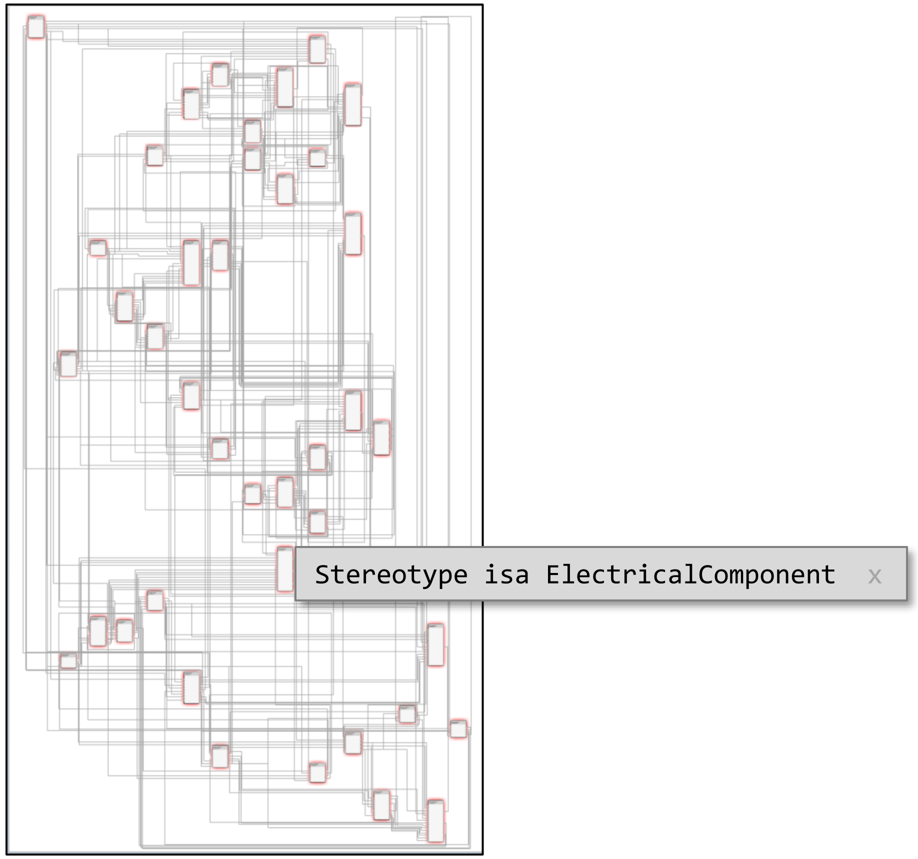
Implemented by:

Drive System

Comments

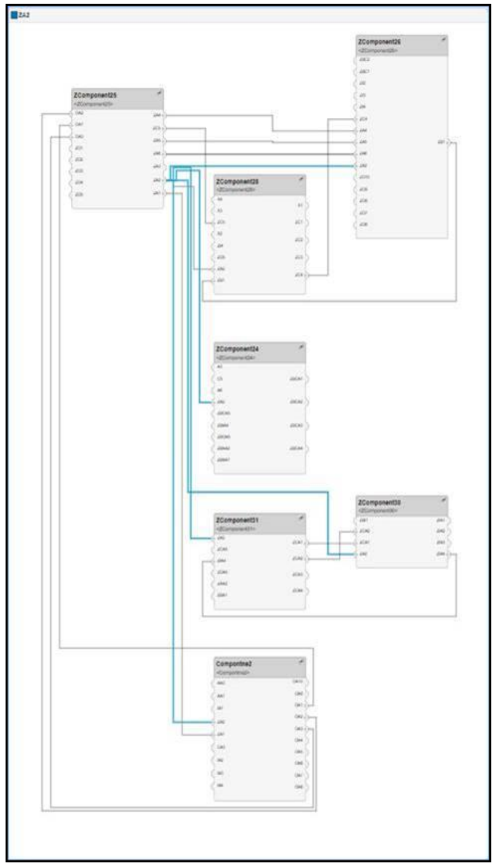
Requirements Toolbox™

Simplify the complex with Filters and autogenerated Views



```
Stereotype isa ElectricalComponent   x
```

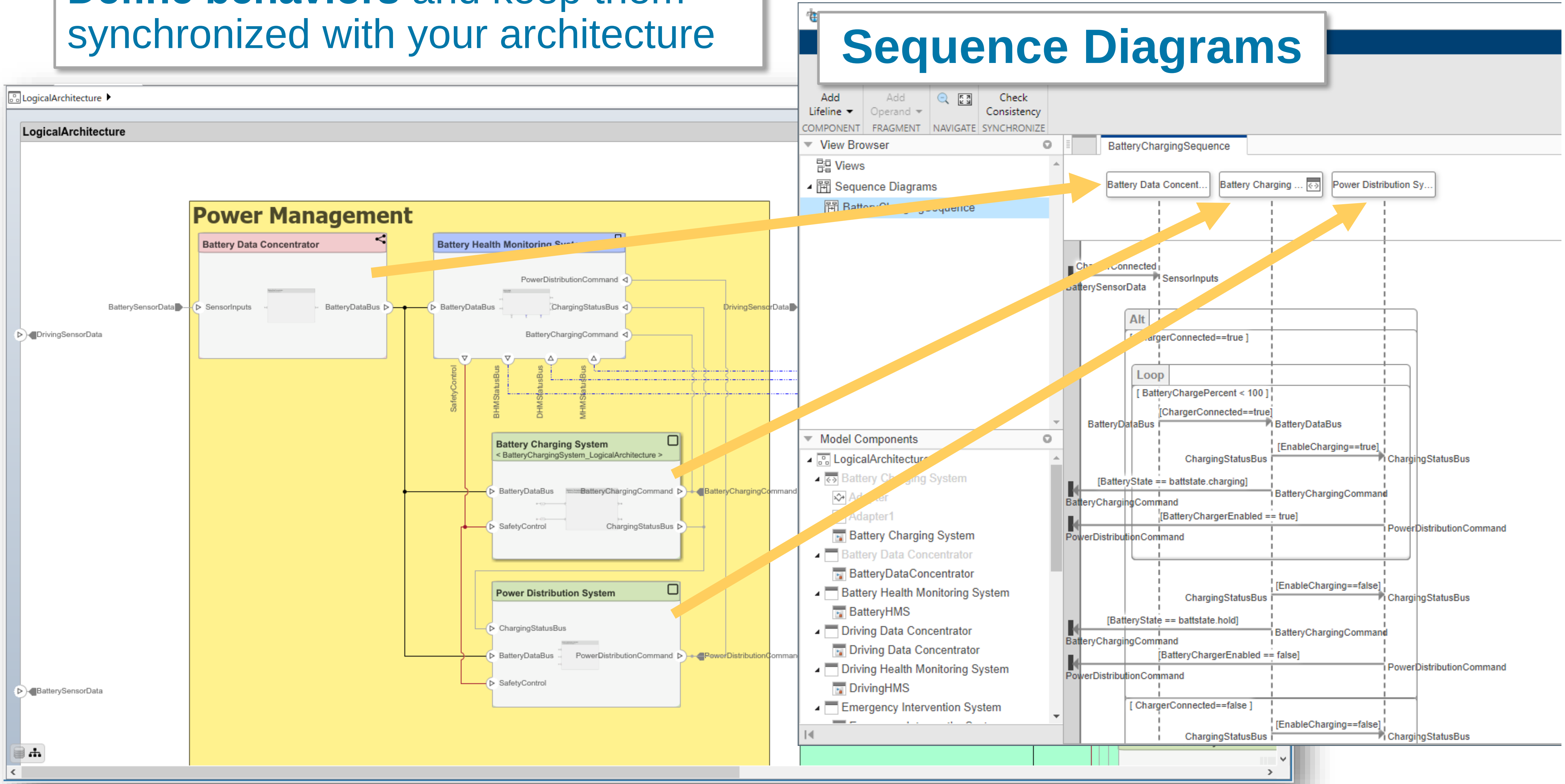
Full system model



Filtered view

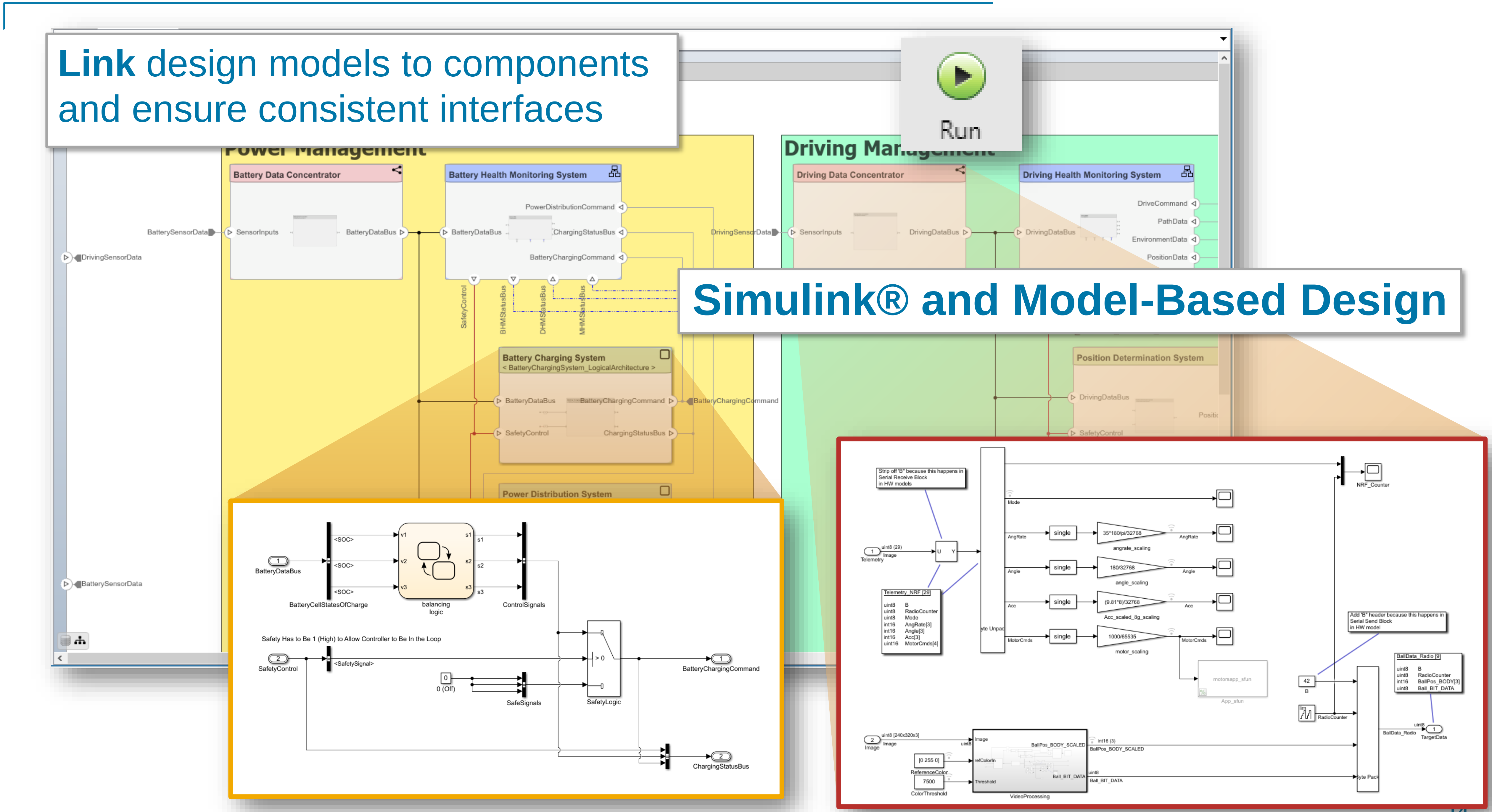
Define behaviors and keep them synchronized with your architecture

Sequence Diagrams

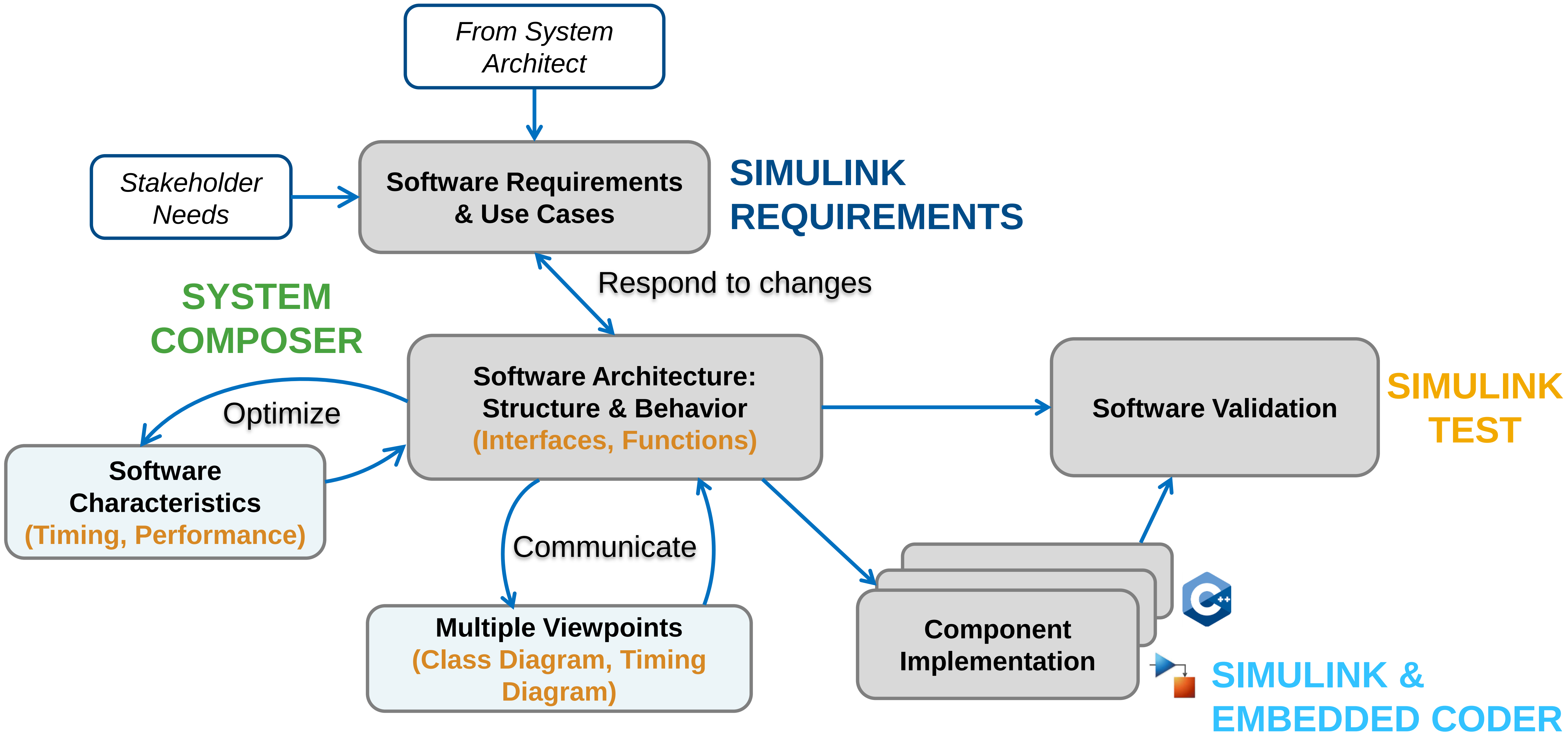


Link design models to components and ensure consistent interfaces

Simulink® and Model-Based Design

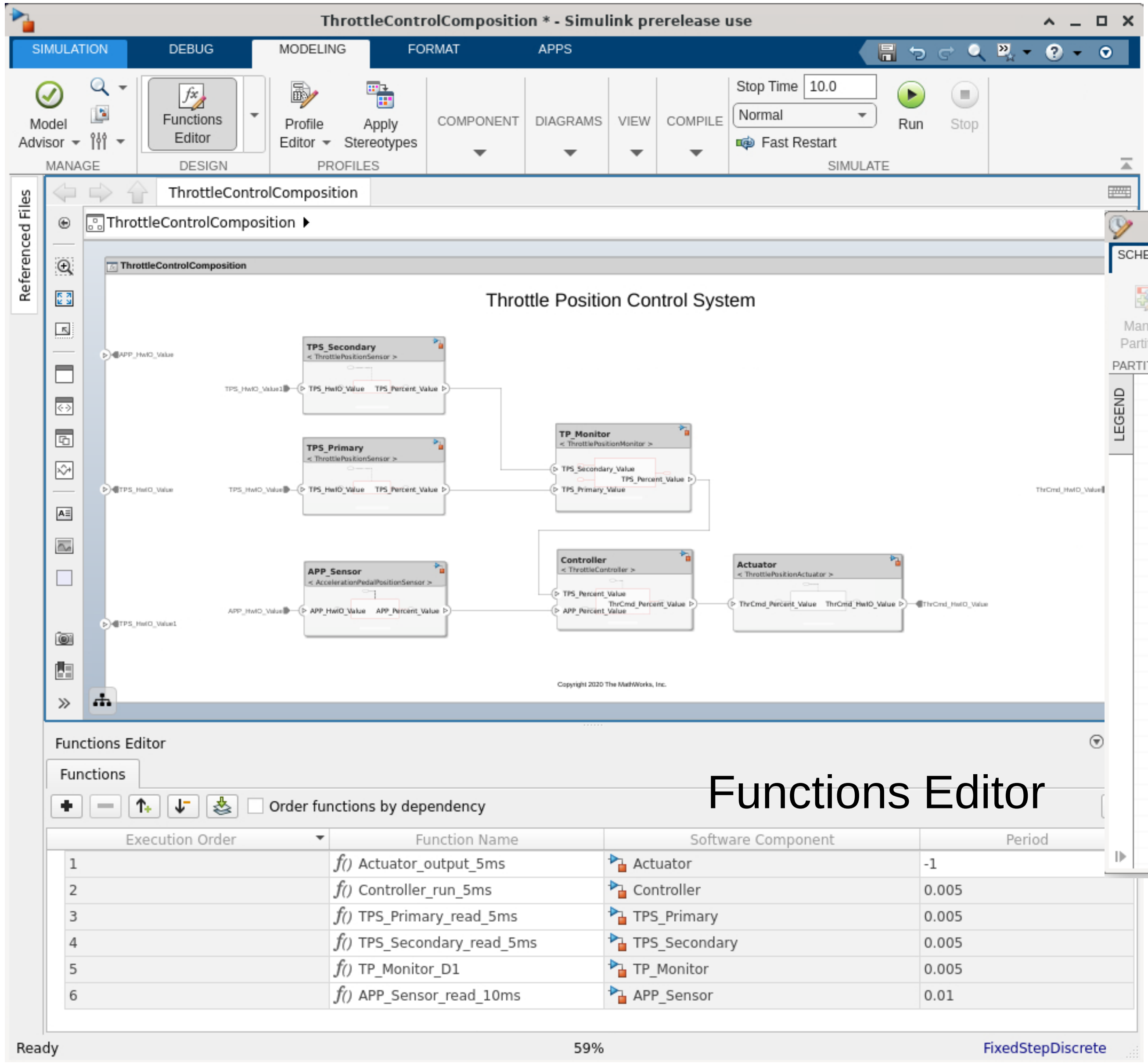


User Workflow for Software Architecture Modeling


From System Architect
Stakeholder Needs
Software Requirements & Use Cases
SIMULINK REQUIREMENTS
Respond to changes
SYSTEM COMPOSER
Software Architecture: Structure & Behavior (Interfaces, Functions)
Optimize
Software Characteristics (Timing, Performance)
Software Validation
SIMULINK TEST
Communicate
Multiple Viewpoints (Class Diagram, Timing Diagram)
Component Implementation
SIMULINK & EMBEDDED CODER

R2021b

Represent Internal Behavior of Components as Functions



Stop Time 10.0
Normal
Fast Restart

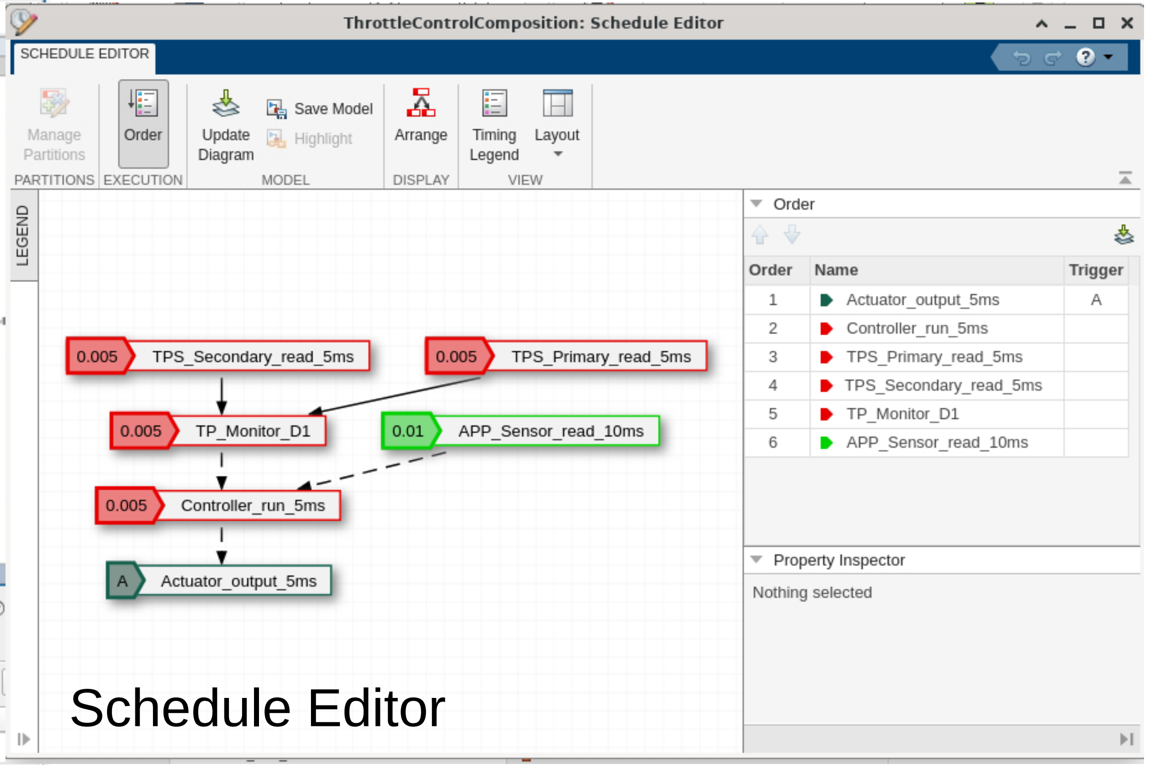
Functions Editor

Functions

Order functions by dependency

Execution Order	Function Name	Software Component	Period
1	f() Actuator_output_5ms	Actuator	-1
2	f() Controller_run_5ms	Controller	0.005
3	f() TPS_Primary_read_5ms	TPS_Primary	0.005
4	f() TPS_Secondary_read_5ms	TPS_Secondary	0.005
5	f() TP_Monitor_D1	TP_Monitor	0.005
6	f() APP_Sensor_read_10ms	APP_Sensor	0.01

Functions Editor



Order

Order	Name	Trigger
1	Actuator_output_5ms	A
2	Controller_run_5ms	
3	TPS_Primary_read_5ms	
4	TPS_Secondary_read_5ms	
5	TP_Monitor_D1	
6	APP_Sensor_read_10ms	

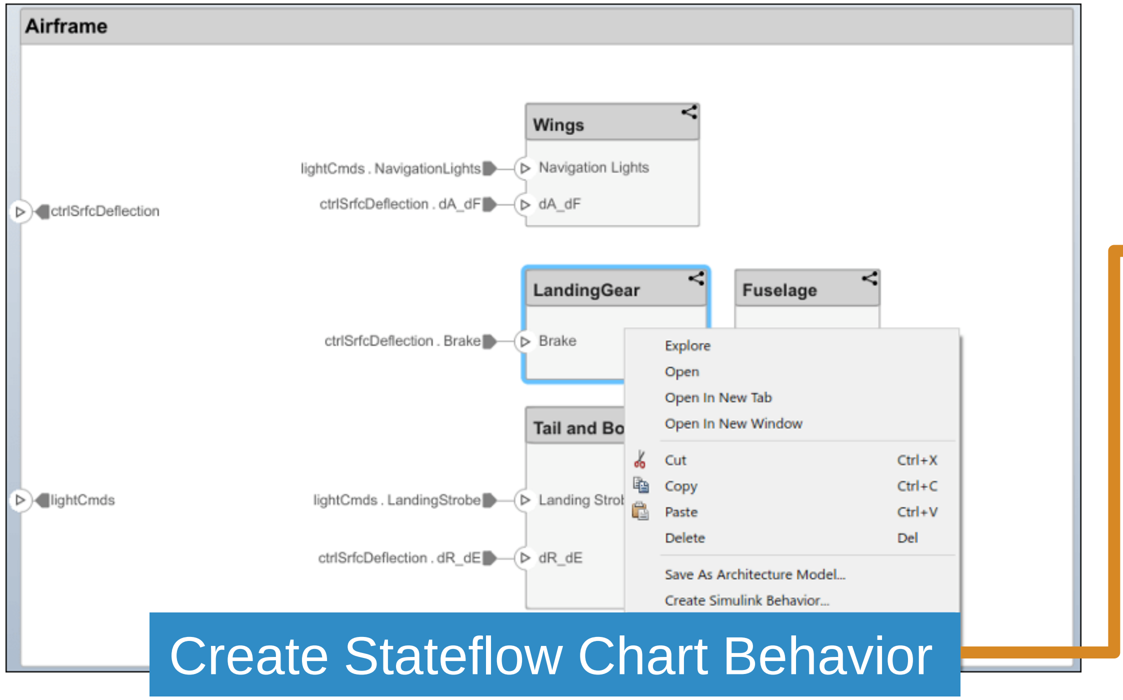
Property Inspector

Nothing selected

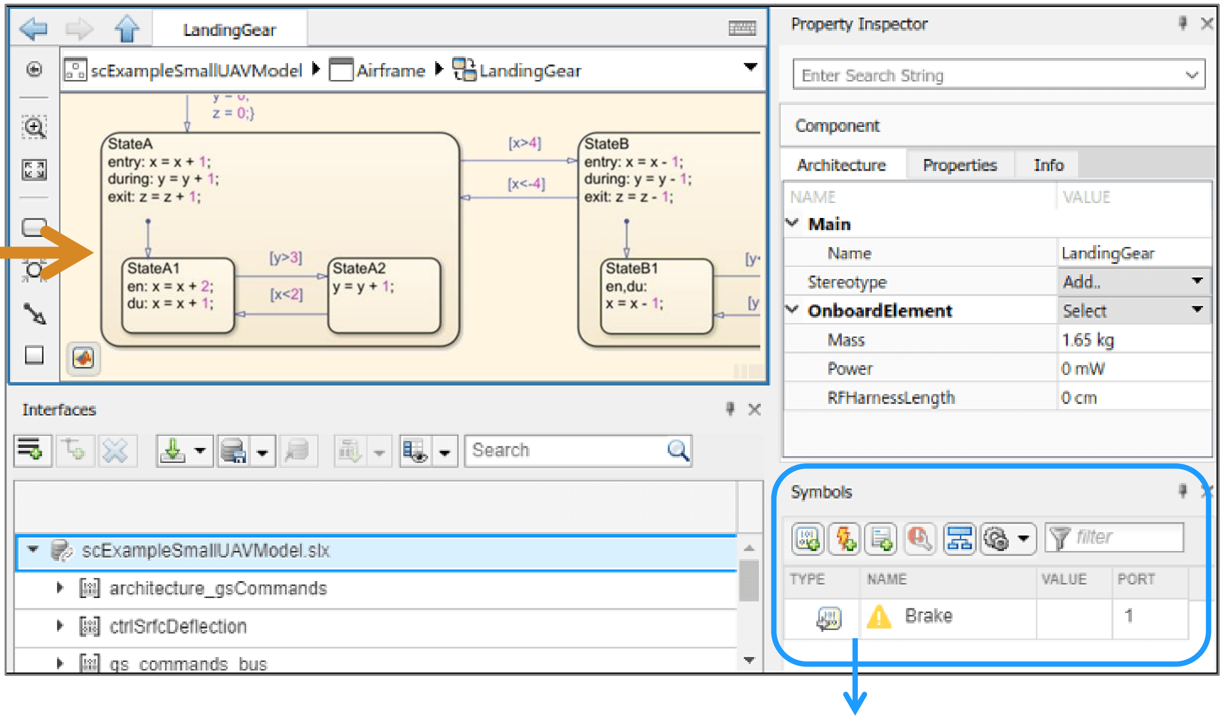
Schedule Editor

Define behaviors and keep them synchronized with your architecture

State Charts



Create Stateflow Chart Behavior



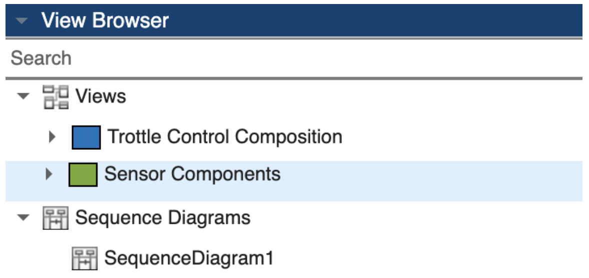
NAME	VALUE
Main	
Name	LandingGear
Stereotype	Add..
OnboardElement	Select
Mass	1.65 kg
Power	0 mW
RFHarnessLength	0 cm

TYPE	NAME	VALUE	PORT
	Brake		1

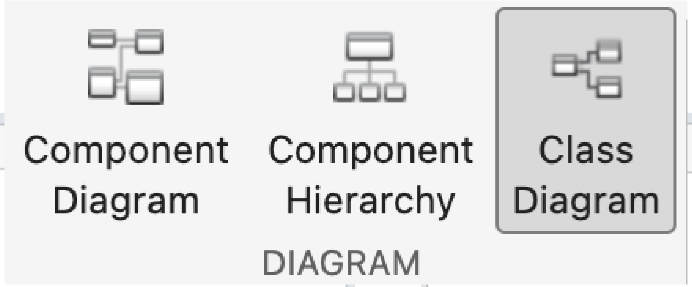
State chart with pre-configured interface

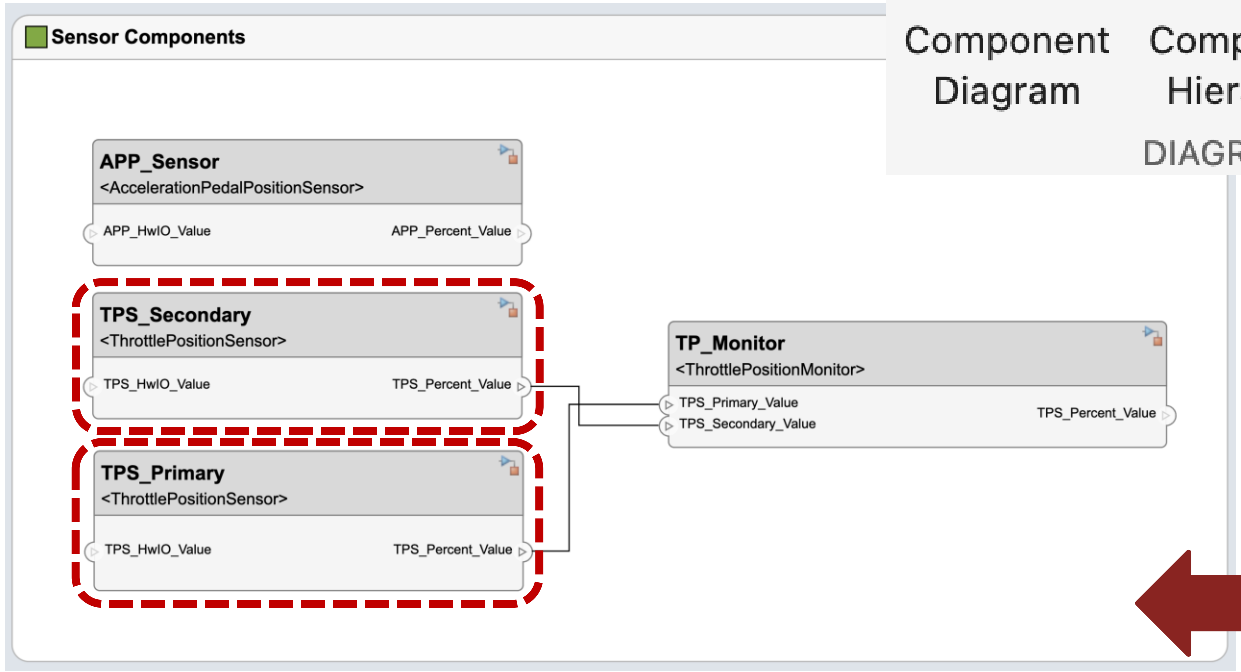
R2021b

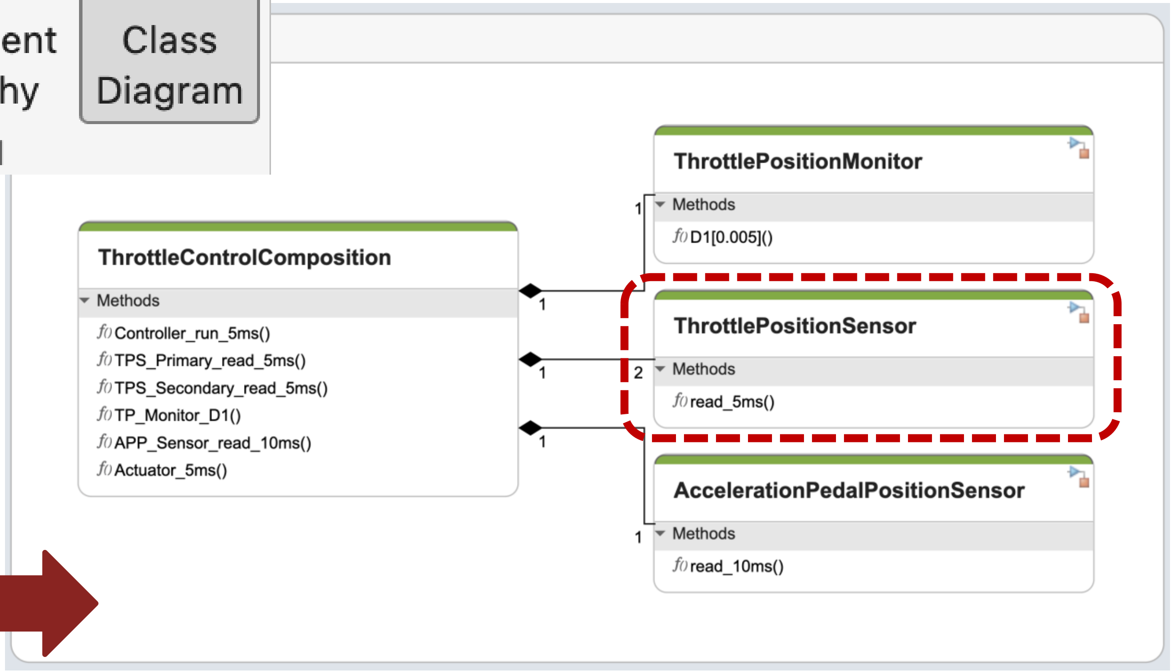
Class Diagram shows unique types of components



“How many throttle position sensors do I have in my software?”

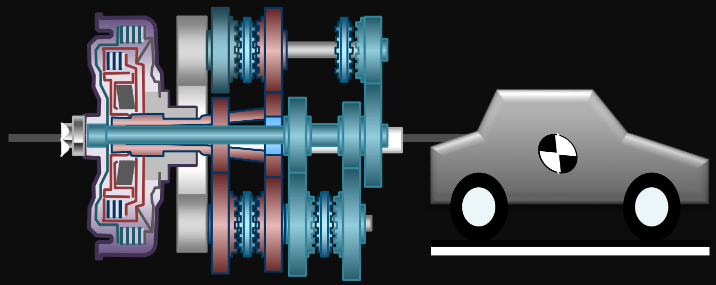






Trending to Service-Oriented Architecture (SOA)

Base Vehicle

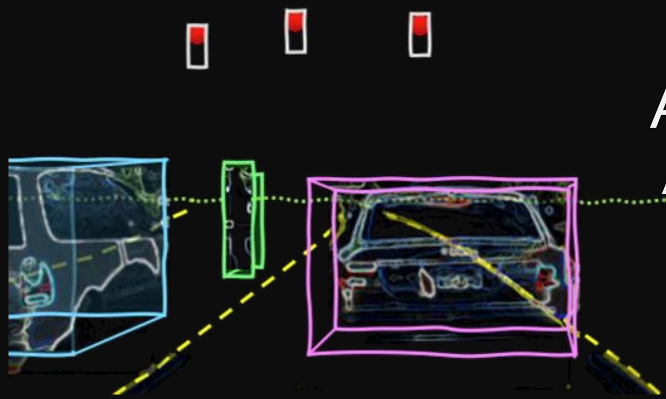


Controls
Real-time
CAN

Speed,
Velocity

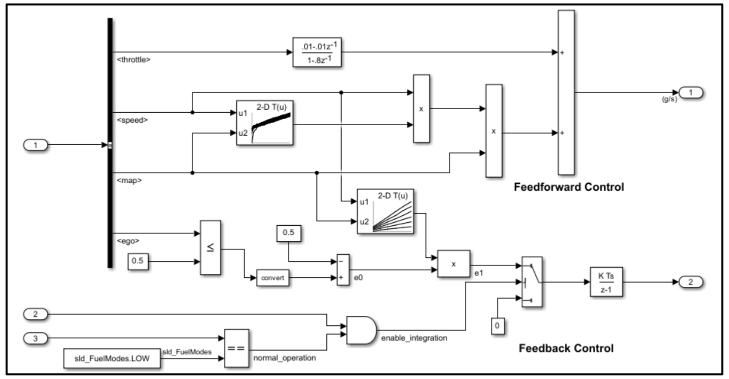
Steering,
Braking

Software-Defined Vehicle



Autonomy
AI / ECUs
Lidar, Camera,…
Cloud

Service-Oriented Architectures



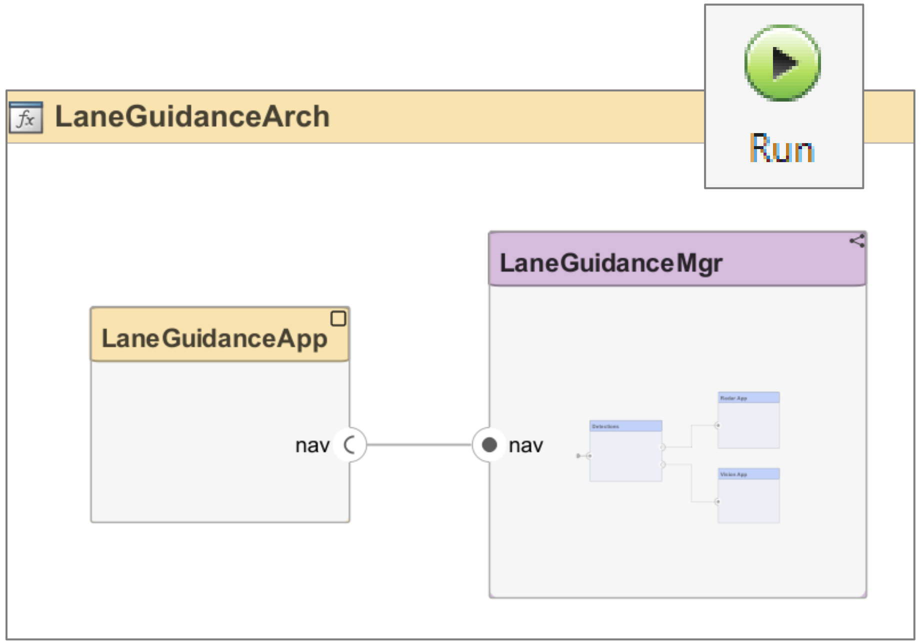
Model-Based Design

+

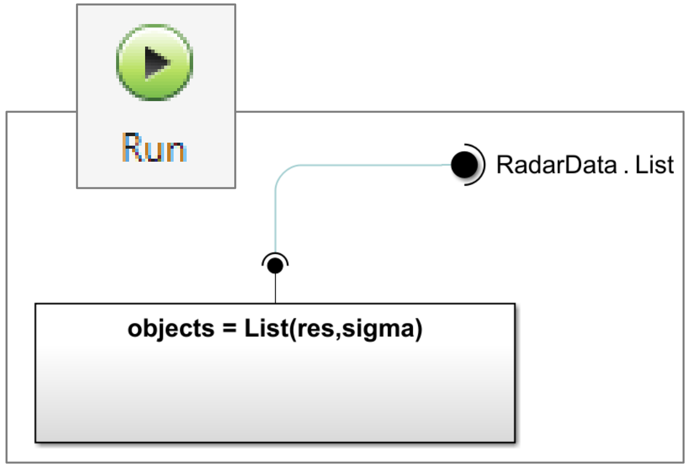
Linux
FACE
Future Airborne Capability Environment
DDS
AUTOSAR Adaptive Platform
ROS

Microservice + service interface

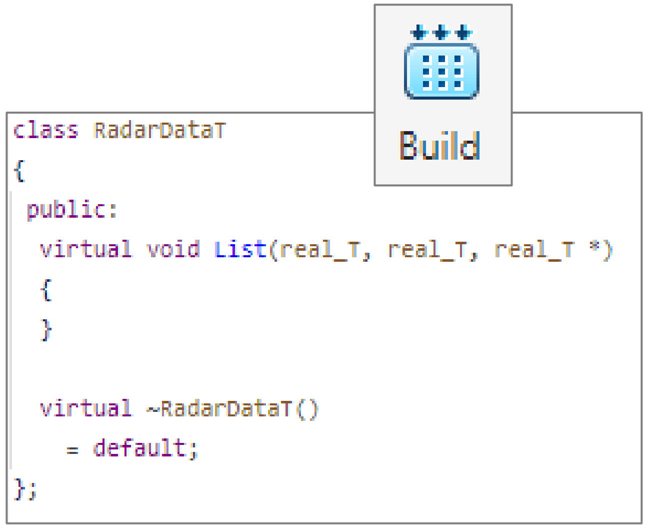
Service-Oriented Architecture (SOA) Design



Describe SOA with **System Composer**



Implement detailed components with **Simulink**



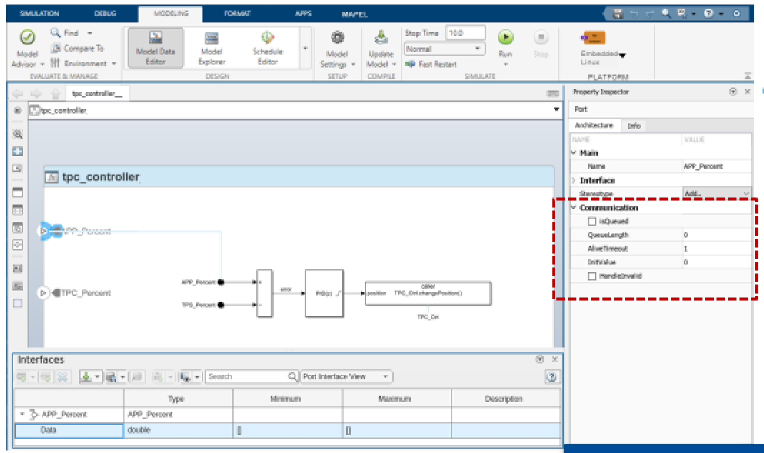
```
class RadarDataT
{
 public:
  virtual void List(real_T, real_T, real_T *)
  {
  }

  virtual ~RadarDataT()
    = default;
};
```

Generate code with **Embedded Coder**

COMPLETE SOA WORKFLOWS

FUTURE RELEASE

Specify and Simulate QoS Communication Attributes that Exist for All Platforms

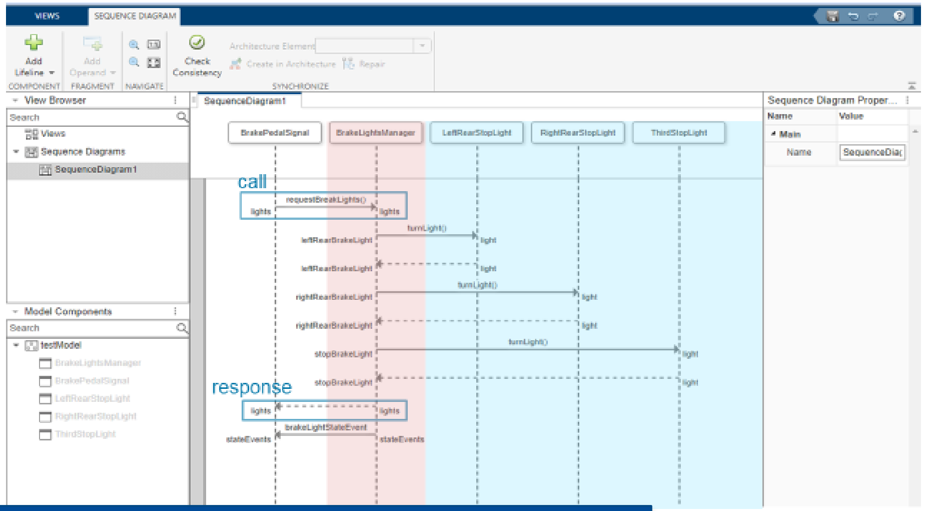


Interface Editor

- Port elements have universal attributes that exist for all platforms
  - Queue properties
  - Error status
- Attributes extensible with stereotypes

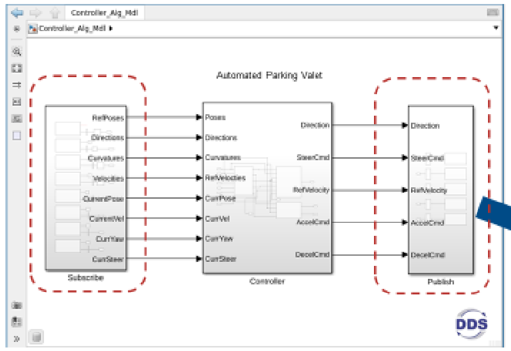
FUTURE RELEASE

Sequence Diagram for Client-Server

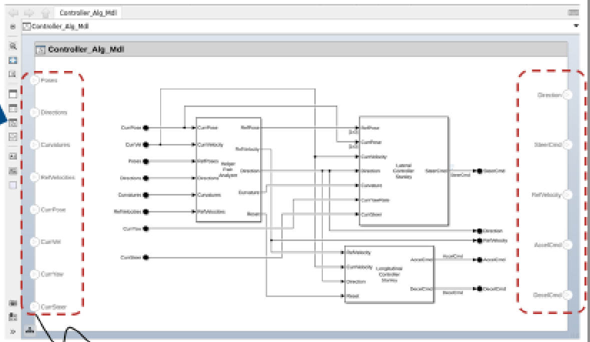


FUTURE RELEASE

Adapt Algorithmic Models to Architecture Interfaces



Algorithm Model with Message Blocks

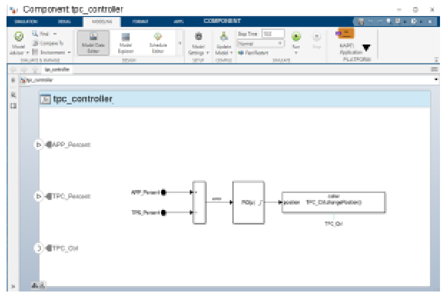


Component Perspective

FUTURE RELEASE

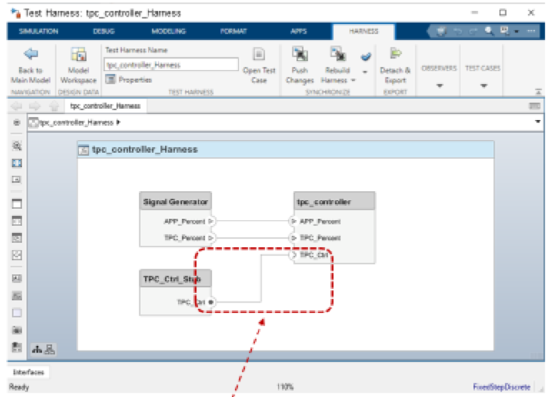
Test Harness Workflow Creates Software Compositions to Simu a Component

COMPONENT



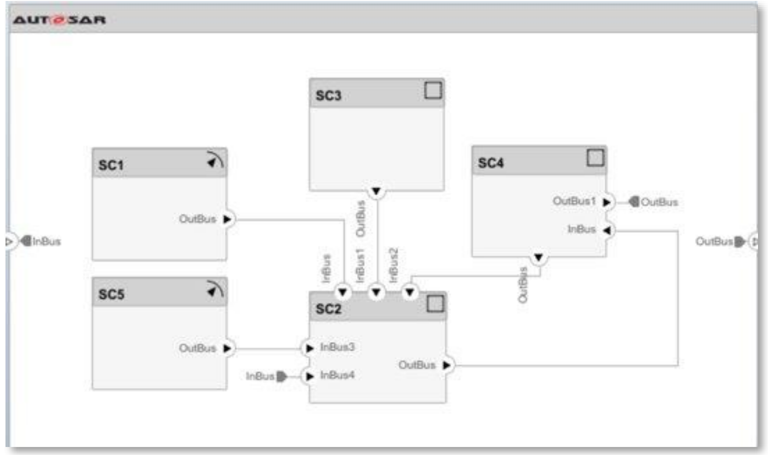
Add test harness

TEST HARNESS



Stub server functions to test client calls

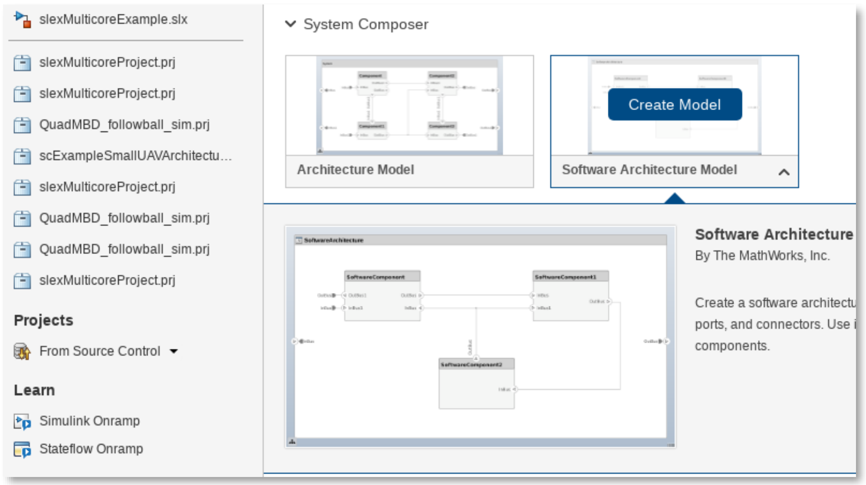
AUTOSAR Architecture



R2019b

- Strong support for Classic
- Growing support for Adaptive

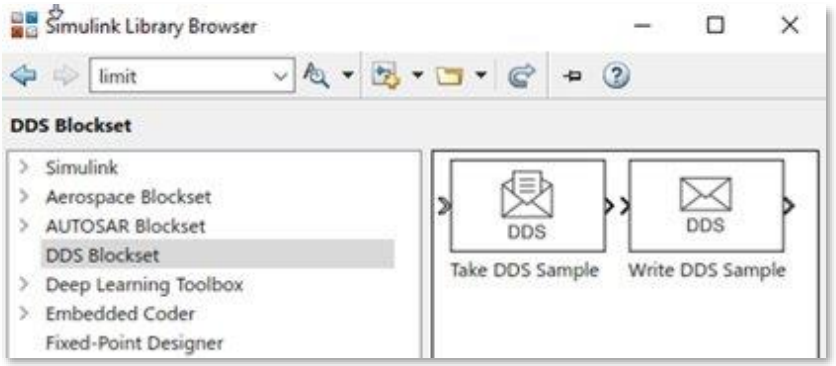
Software Architecture


slexMulticoreExample.slx
slexMulticoreProject.prj
slexMulticoreProject.prj
QuadMBD_followball_sim.prj
scExampleSmallUAVArchitectu...
slexMulticoreProject.prj
QuadMBD_followball_sim.prj
QuadMBD_followball_sim.prj
slexMulticoreProject.prj
Projects
From Source Control
Learn
Simulink Onramp
Stateflow Onramp
System Composer
Architecture Model
Software Architecture Model
Create Model
Software Architecture
By The MathWorks, Inc.
Create a software architectu ports, and connectors. Use i components.

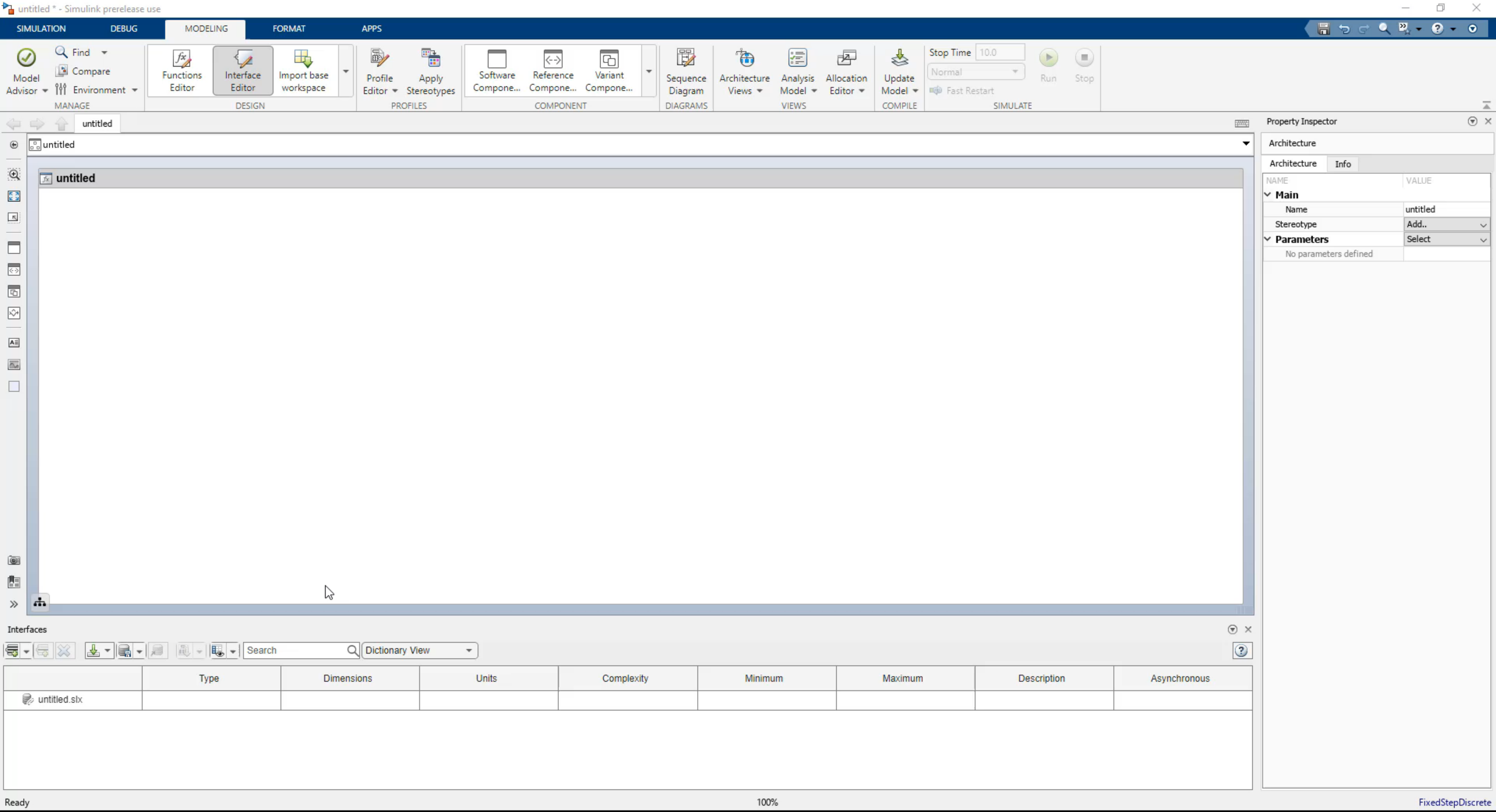
R2021a

- Embedded coder support package for Linux emerging

DDS Application


Simulink Library Browser
limit
DDS Blockset
Simulink
Aerospace Blockset
AUTOSAR Blockset
DDS Blockset
Deep Learning Toolbox
Embedded Coder
Fixed-Point Designer
DDS
Take DDS Sample
DDS
Write DDS Sample

R2021a



Stop Time 10.0

Normal

Property Inspector

Architecture

Architecture | Info

NAME	VALUE
Main	
Name	untitled
Stereotype	Add..
Parameters	Select
No parameters defined	

	Type	Dimensions	Units	Complexity	Minimum	Maximum	Description	Asynchronous
untitled.slx								

Towards Virtual Integration and Simulation

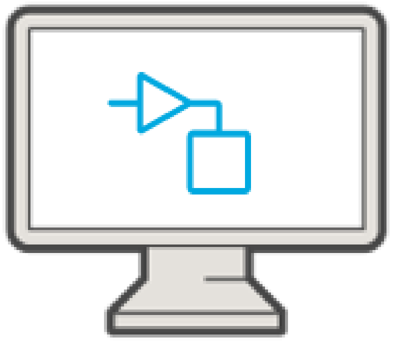
Shifting Left: How far can you go?

Verify that the integration of Application SW components into full Application meets functional requirements

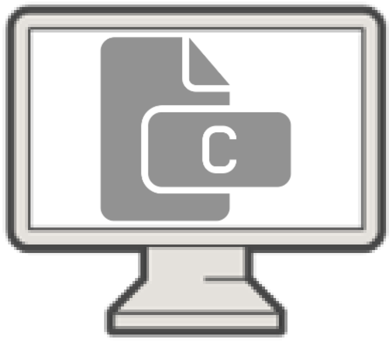
Verify the integration of Application SW with Basic SW

Validate the integration of one/few ECUs with simulated or real sensors, actuators, networks

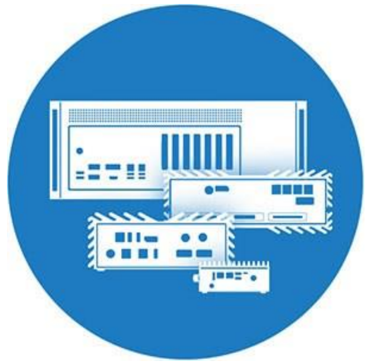
Validate the integration of **ALL** ECUs, Networks, Sensors and Actuators



MIL



SIL

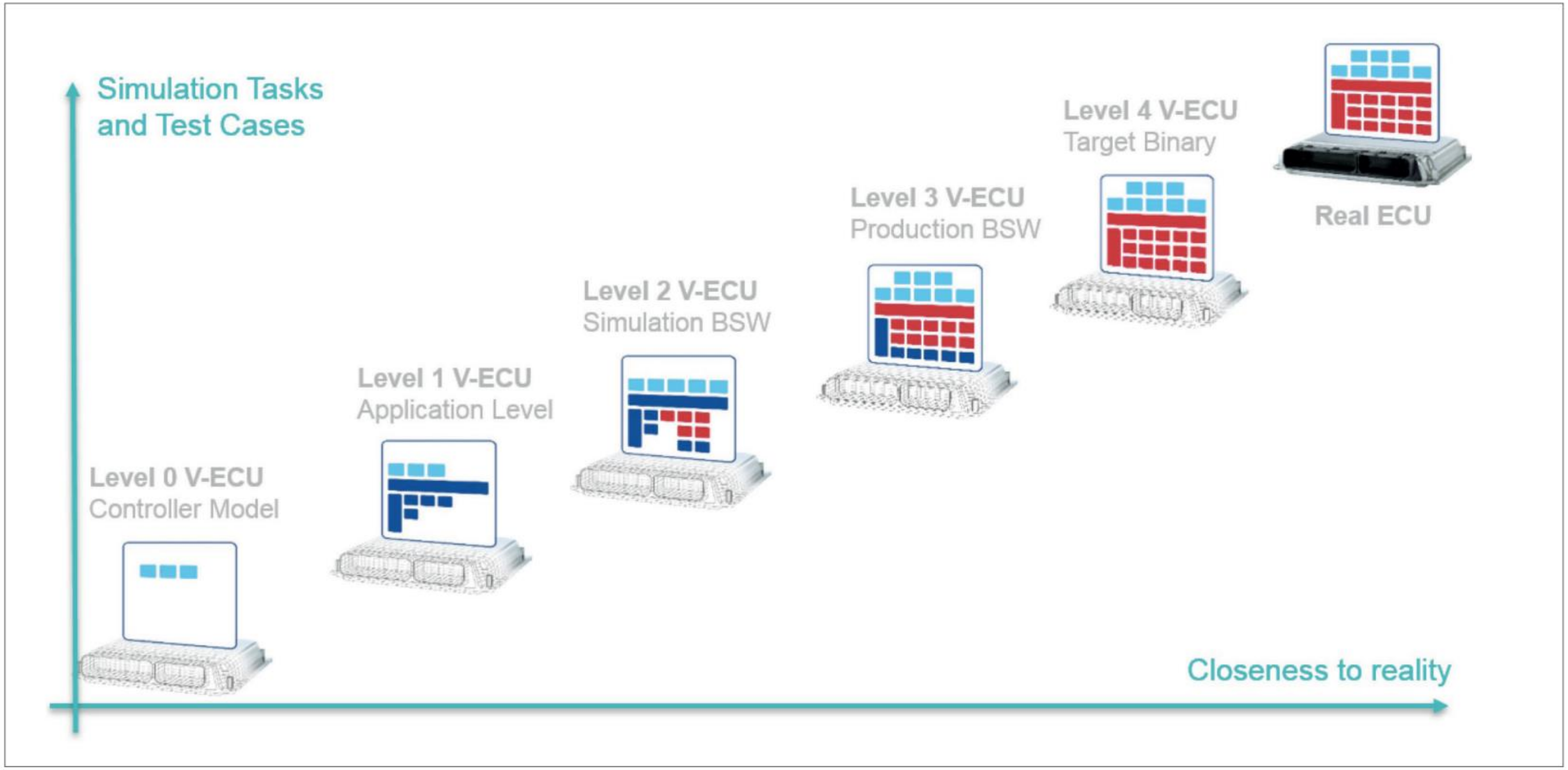


HIL

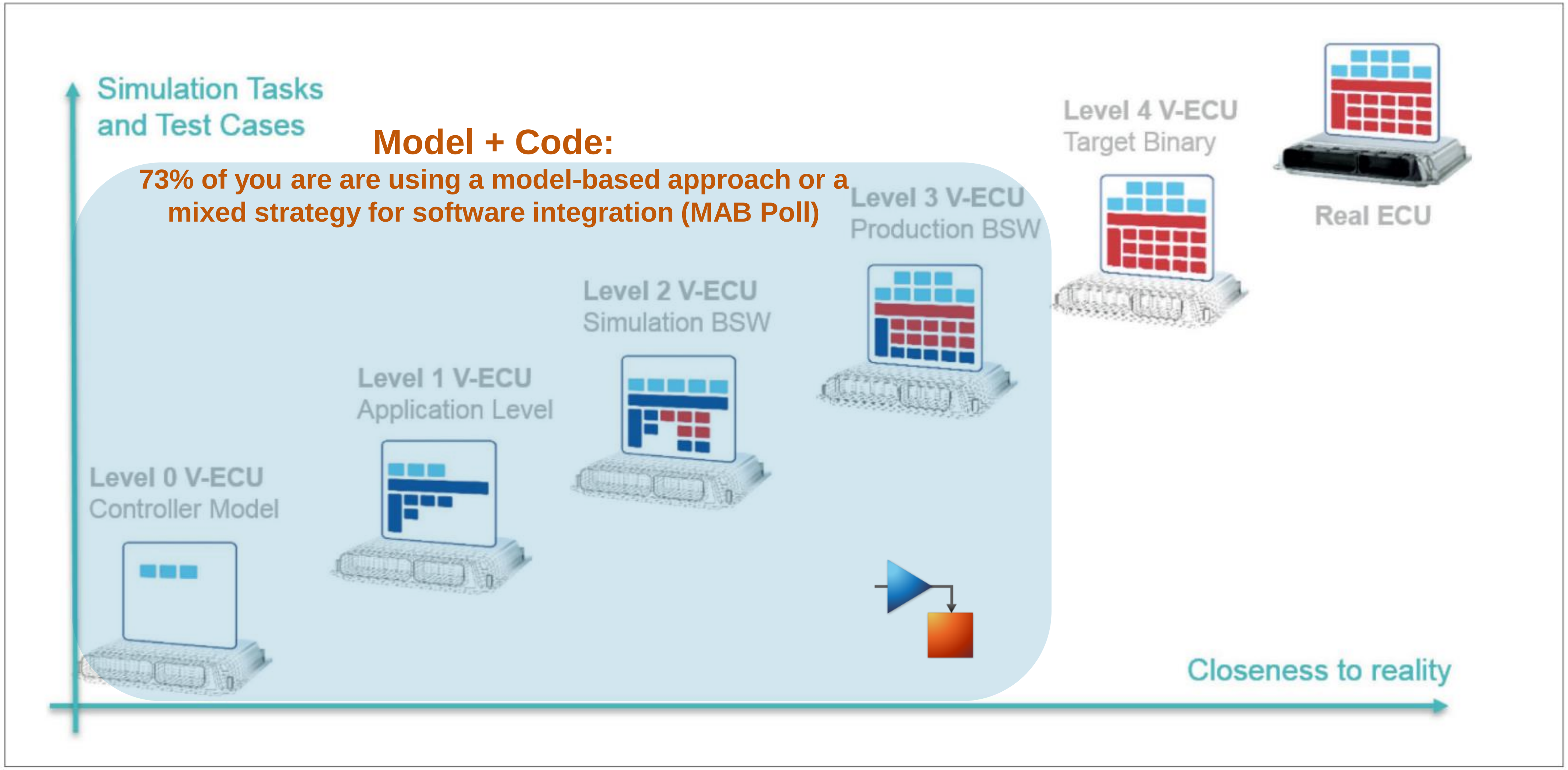
Vehicle

Key takeaway: use each test facility where it adds value during the process

V-ECU: Use cases vs level of fidelity

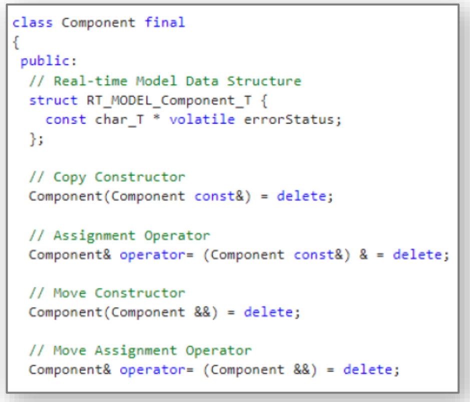


V-ECU with Simulink: Focus on where the complexity grows



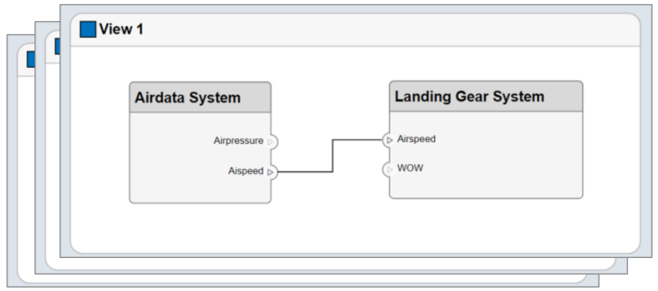
Core capabilities we are focusing on

“Ready-to-run” Model Components



```
class Component final
{
 public:
  // Real-time Model Data Structure
  struct RT_MODEL_Component_T {
    const char_T * volatile errorStatus;
  };

  // Copy Constructor
  Component(Component const&) = delete;

  // Assignment Operator
  Component& operator= (Component const&) & = delete;

  // Move Constructor
  Component(Component &&) = delete;

  // Move Assignment Operator
  Component& operator= (Component &&) = delete;
```

“Ready-to-run” Code Components



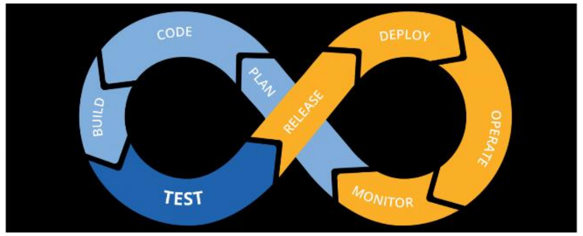
Automated assembly of models

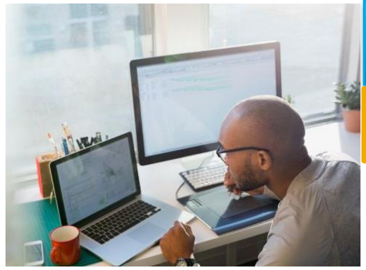
Performant Simulation!

Protected Models evolve to *Ready-to-run* models for integration

FUTURE RELEASE

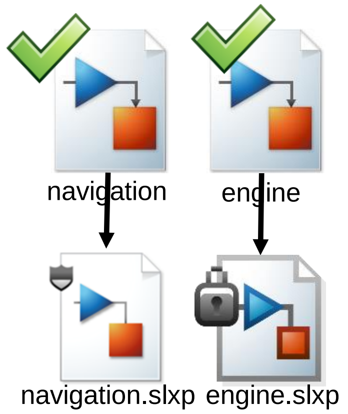
Creator



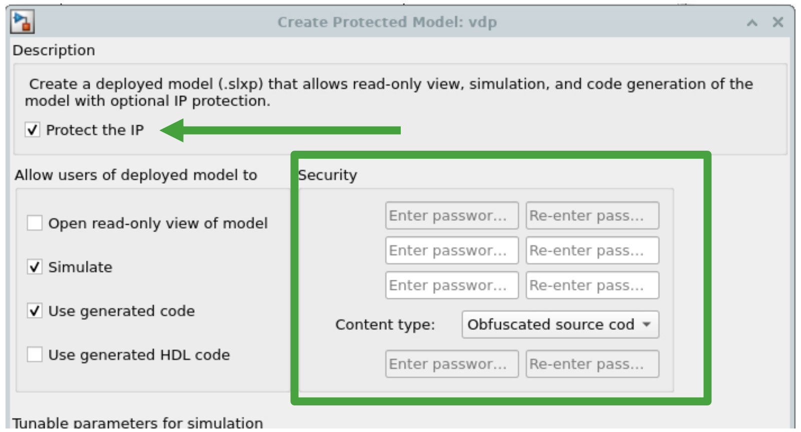


Recipient

Inside/outside the org



Package **verified** component for ready-to-run



Create Protected Model: vdp

Description

Create a deployed model (.slxp) that allows read-only view, simulation, and code generation of the model with optional IP protection.

☑ Protect the IP

Allow users of deployed model to

☐ Open read-only view of model

☑ Simulate

☑ Use generated code

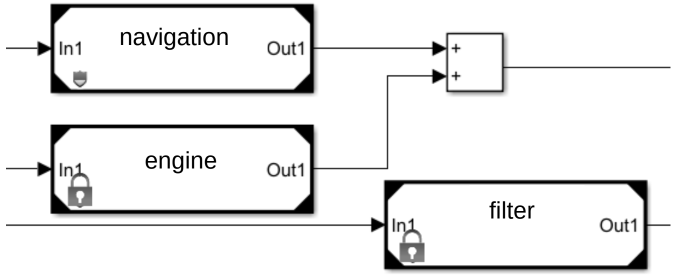
☐ Use generated HDL code

Security

Enter passwor... Re-enter pass...

Content type: Obfuscated source cod

Tunable parameters for simulation

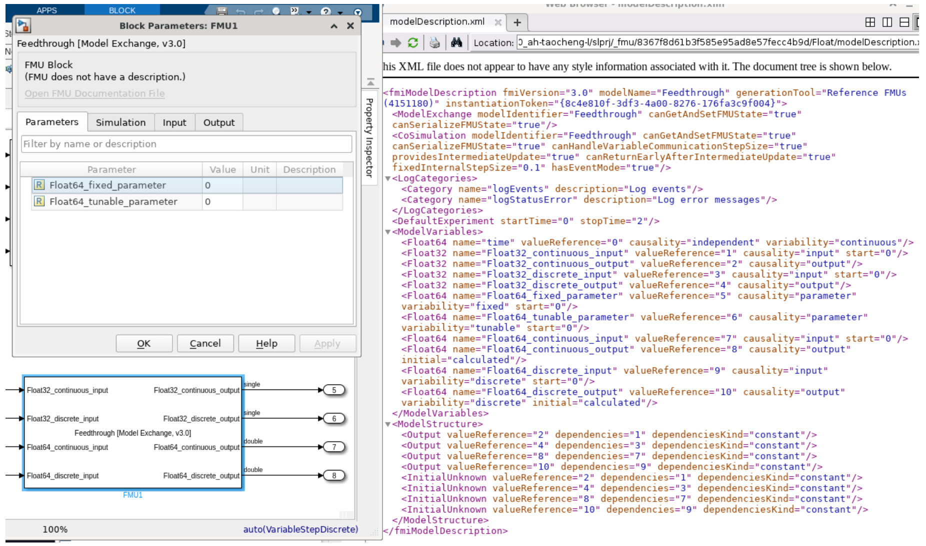


Benefits

- Flexibility
- Performance
- Encapsulation
- Multi-instantiation
- …

FMUs continue to provide an avenue to make ready-to-run Parts from other tools

- Simulink supports FMI 3.0 Import in R2023b



Block Parameters: FMU1

Feedthrough [Model Exchange, v3.0]

FMU Block
(FMU does not have a description.)
Open FMU Documentation File

Parameters | Simulation | Input | Output

Filter by name or description

Parameter	Value	Unit	Description
Float64_fixed_parameter	0		
Float64_tunable_parameter	0		

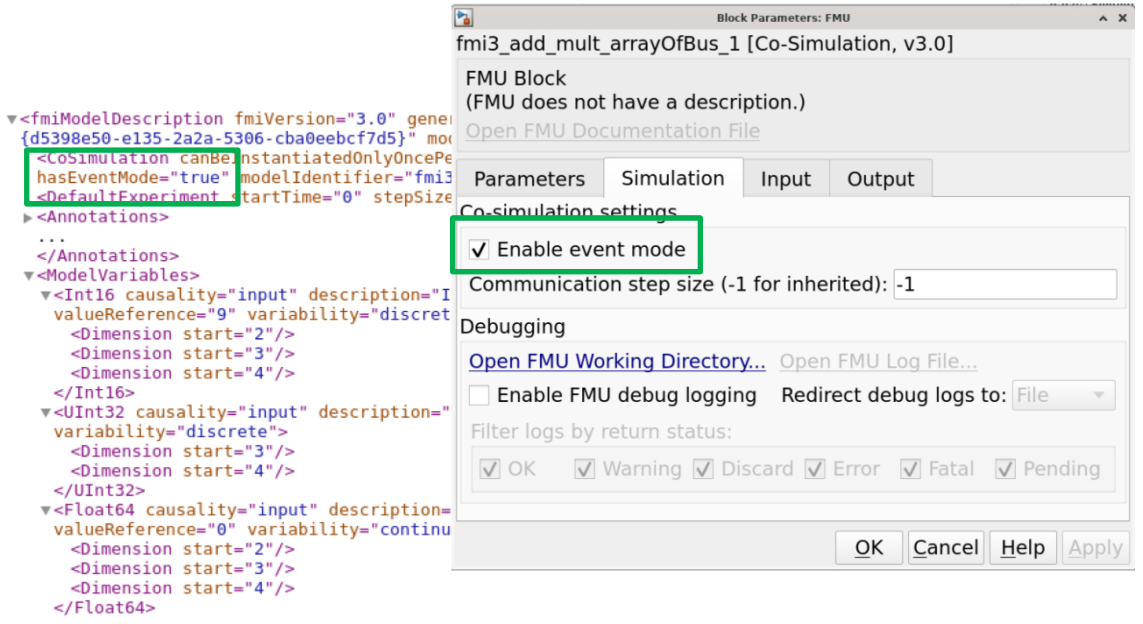
OK | Cancel | Help | Apply

Location: 0_ah-taocheng-l/slprj/_fmu/8367f8d61b3f585e95ad8e57fecc4b9d/Float/modelDescription.

his XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<fmiModelDescription fmiVersion="3.0" modelName="Feedthrough" generationTool="Reference FMUs
(4151180)" instantiationToken="{8c4e810f-3df3-4a00-8276-176fa3c9f004}">
  <ModelExchange modelIdentifier="Feedthrough" canGetAndSetFMUState="true"
  canSerializeFMUState="true"/>
  <CoSimulation modelIdentifier="Feedthrough" canGetAndSetFMUState="true"
  canSerializeFMUState="true" canHandleVariableCommunicationStepSize="true"
  providesIntermediateUpdate="true" canReturnEarlyAfterIntermediateUpdate="true"
  fixedInternalStepSize="0.1" hasEventMode="true"/>
▼<LogCategories>
    <Category name="logEvents" description="Log events"/>
    <Category name="logStatusError" description="Log error messages"/>
  </LogCategories>
  <DefaultExperiment startTime="0" stopTime="2"/>
▼<ModelVariables>
    <Float64 name="time" valueReference="0" causality="independent" variability="continuous"/>
    <Float32 name="Float32_continuous_input" valueReference="1" causality="input" start="0"/>
    <Float32 name="Float32_continuous_output" valueReference="2" causality="output"/>
    <Float32 name="Float32_discrete_input" valueReference="3" causality="input" start="0"/>
    <Float32 name="Float32_discrete_output" valueReference="4" causality="output"/>
    <Float64 name="Float64_fixed_parameter" valueReference="5" causality="parameter"
    variability="fixed" start="0"/>
    <Float64 name="Float64_tunable_parameter" valueReference="6" causality="parameter"
    variability="tunable" start="0"/>
    <Float64 name="Float64_continuous_input" valueReference="7" causality="input" start="0"/>
    <Float64 name="Float64_continuous_output" valueReference="8" causality="output"
    initial="calculated"/>
    <Float64 name="Float64_discrete_input" valueReference="9" causality="input"
    variability="discrete" start="0"/>
    <Float64 name="Float64_discrete_output" valueReference="10" causality="output"
    variability="discrete" initial="calculated"/>
  </ModelVariables>
▼<ModelStructure>
    <Output valueReference="2" dependencies="1" dependenciesKind="constant"/>
    <Output valueReference="4" dependencies="3" dependenciesKind="constant"/>
    <Output valueReference="8" dependencies="7" dependenciesKind="constant"/>
    <Output valueReference="10" dependencies="9" dependenciesKind="constant"/>
    <InitialUnknown valueReference="2" dependencies="1" dependenciesKind="constant"/>
    <InitialUnknown valueReference="4" dependencies="3" dependenciesKind="constant"/>
    <InitialUnknown valueReference="8" dependencies="7" dependenciesKind="constant"/>
    <InitialUnknown valueReference="10" dependencies="9" dependenciesKind="constant"/>
  </ModelStructure>
</fmiModelDescription>
```

FMU Import block loading FMU 3.0 modelDescription file



```
▼<fmiModelDescription fmiVersion="3.0" gene
 {d5398e50-e135-2a2a-5306-cba0eebcf7d5}" mod
   <CoSimulation canBeInstantiatedOnlyOncePe
   hasEventMode="true" modelIdentifier="fmi3
   <DefaultExperiment startTime="0" stepSize
 ▶<Annotations>
   ...
   </Annotations>
 ▼<ModelVariables>
   ▼<Int16 causality="input" description="I
     valueReference="9" variability="discret
       <Dimension start="2"/>
       <Dimension start="3"/>
       <Dimension start="4"/>
     </Int16>
   ▼<UInt32 causality="input" description="
     variability="discrete">
       <Dimension start="3"/>
       <Dimension start="4"/>
     </UInt32>
   ▼<Float64 causality="input" description=
     valueReference="0" variability="continu
       <Dimension start="2"/>
       <Dimension start="3"/>
       <Dimension start="4"/>
     </Float64>
```

Block Parameters: FMU

fmi3_add_mult_arrayOfBus_1 [Co-Simulation, v3.0]

FMU Block
(FMU does not have a description.)
Open FMU Documentation File

Parameters | Simulation | Input | Output

Co-simulation settings

[✓] Enable event mode

Communication step size (-1 for inherited): -1

Debugging

Open FMU Working Directory... Open FMU Log File...

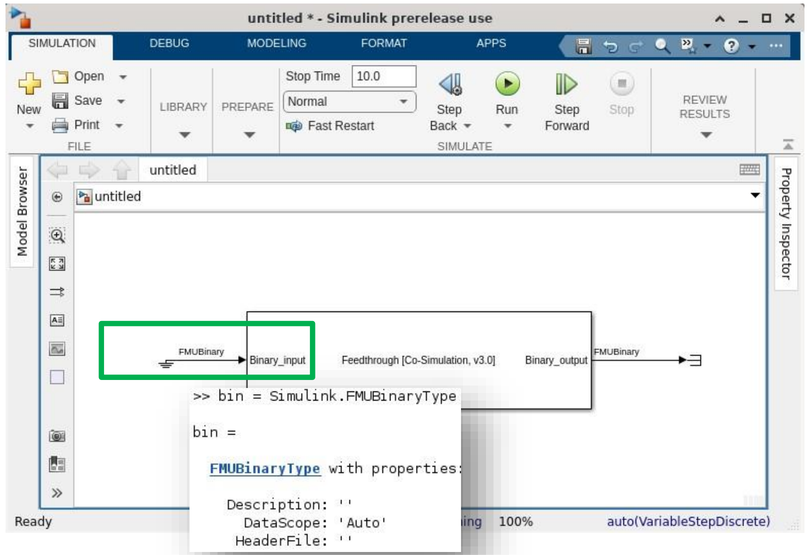
[ ] Enable FMU debug logging   Redirect debug logs to: File

Filter logs by return status:

[✓] OK [✓] Warning [✓] Discard [✓] Error [✓] Fatal [✓] Pending

OK | Cancel | Help | Apply

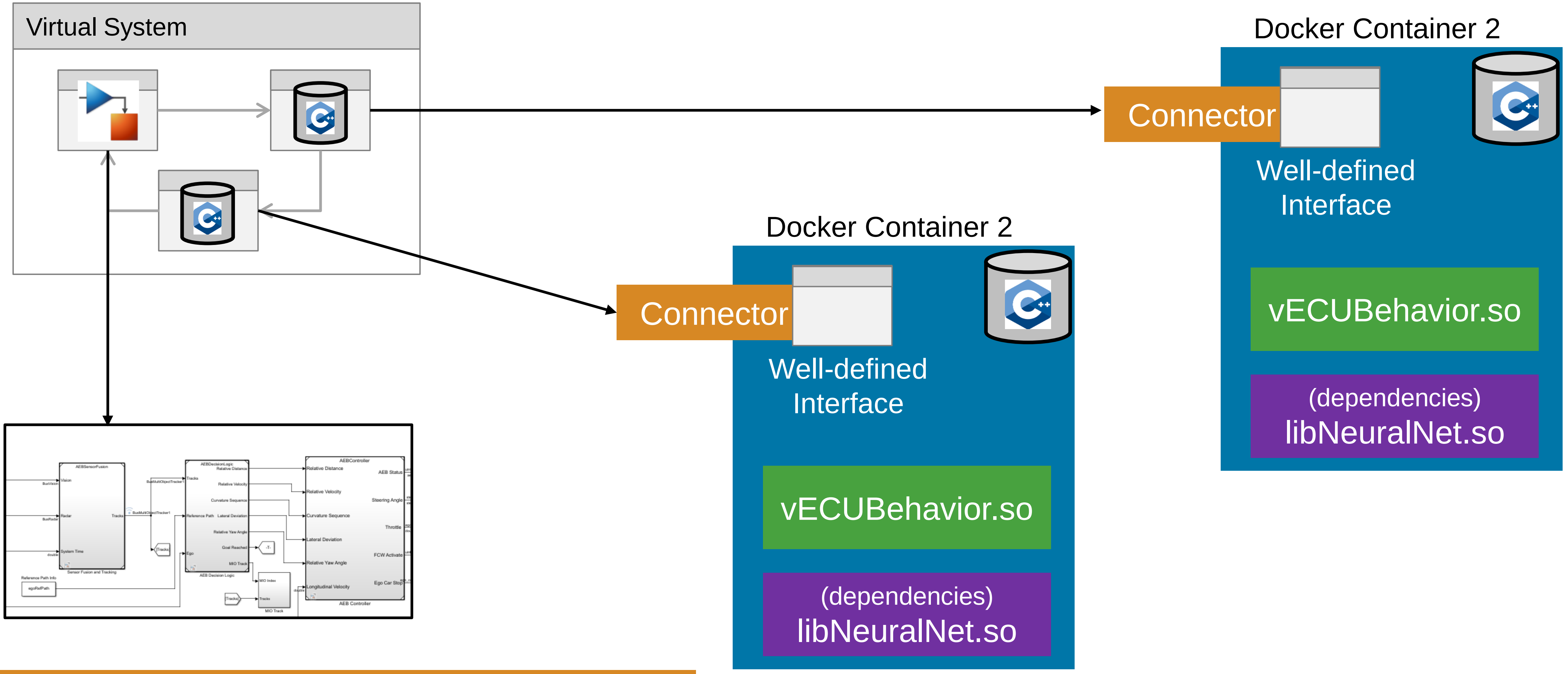
FMU Co-simulation with event mode eliminates one-step delay



```
>> bin = Simulink.FMUBinaryType

bin =

  FMUBinaryType with properties:

     Description: ''
       DataScope: 'Auto'
      HeaderFile: ''
```

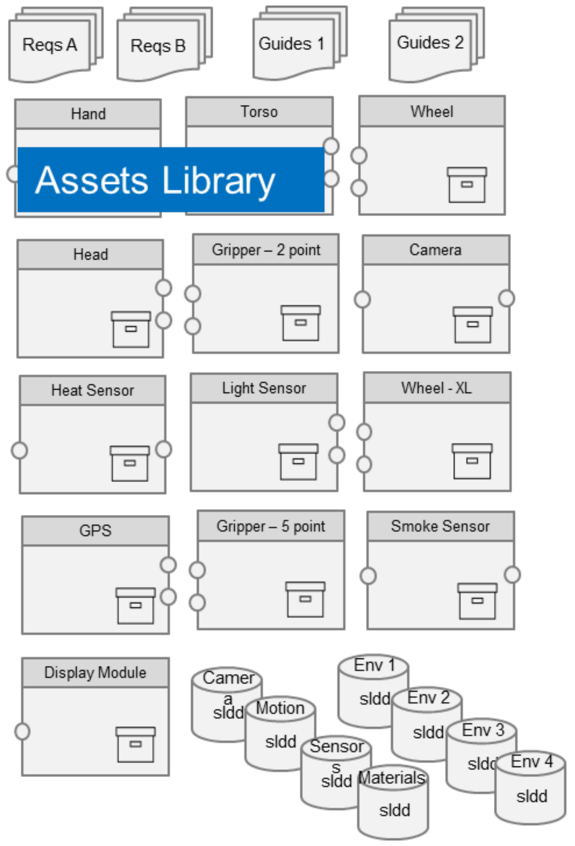
FMUBinary data type

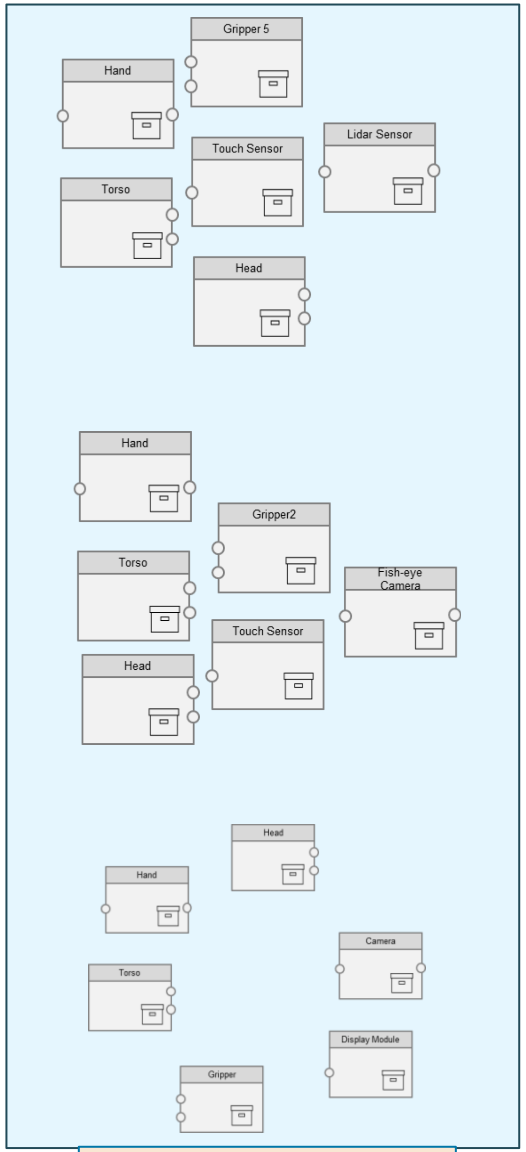
Connector for ready-to-run code components



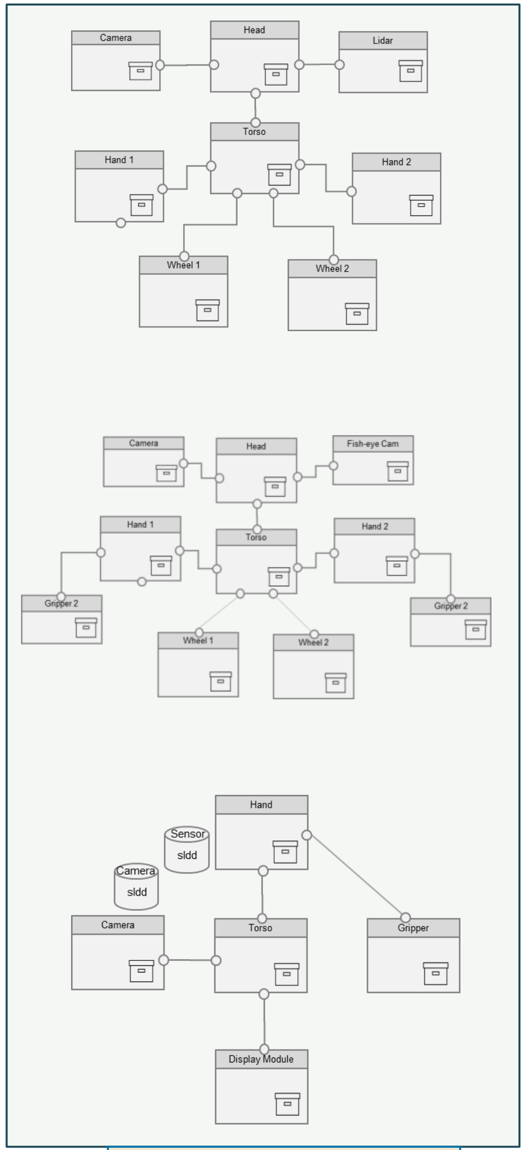
Unit testable, distributable containers

Auto model assembly is being emphasized in many workflows





Selection of Parts



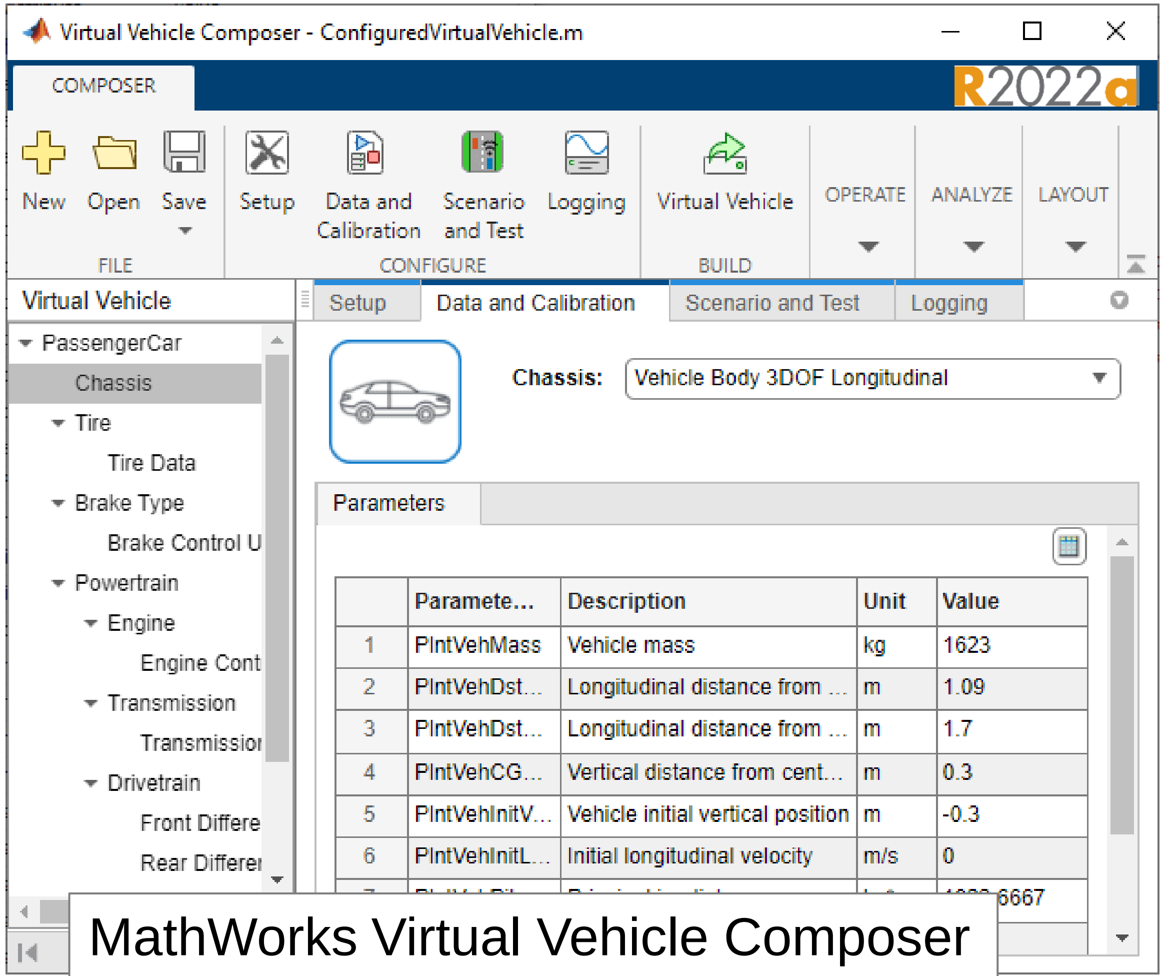
Model Assembly

Explorer

Rescuer

Picker

We are creating a “Feature-Driven” Approach to picking the Parts to Assemble into Models



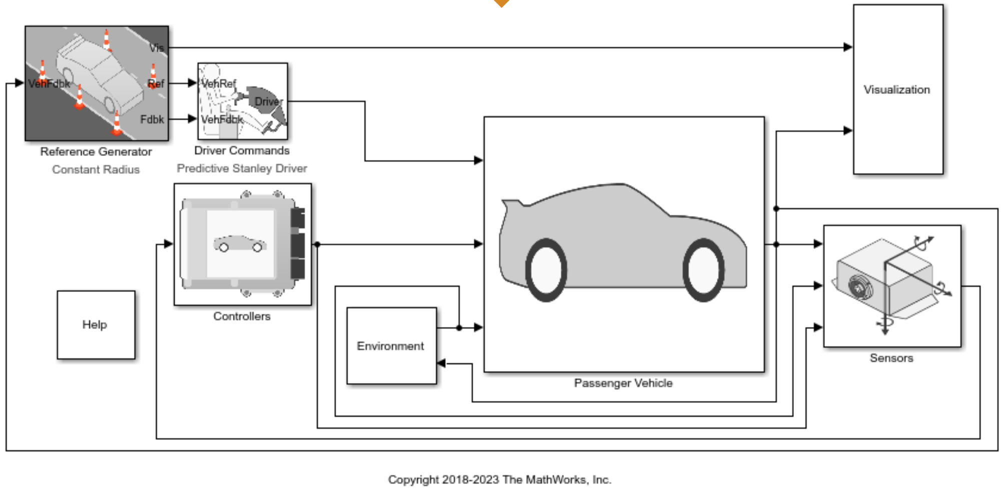
Chassis: Vehicle Body 3DOF Longitudinal

	Paramete...	Description	Unit	Value
1	PlntVehMass	Vehicle mass	kg	1623
2	PlntVehDst...	Longitudinal distance from ...	m	1.09
3	PlntVehDst...	Longitudinal distance from ...	m	1.7
4	PlntVehCG...	Vertical distance from cent...	m	0.3
5	PlntVehInitV...	Vehicle initial vertical position	m	-0.3
6	PlntVehInitL...	Initial longitudinal velocity	m/s	0

MathWorks Virtual Vehicle Composer

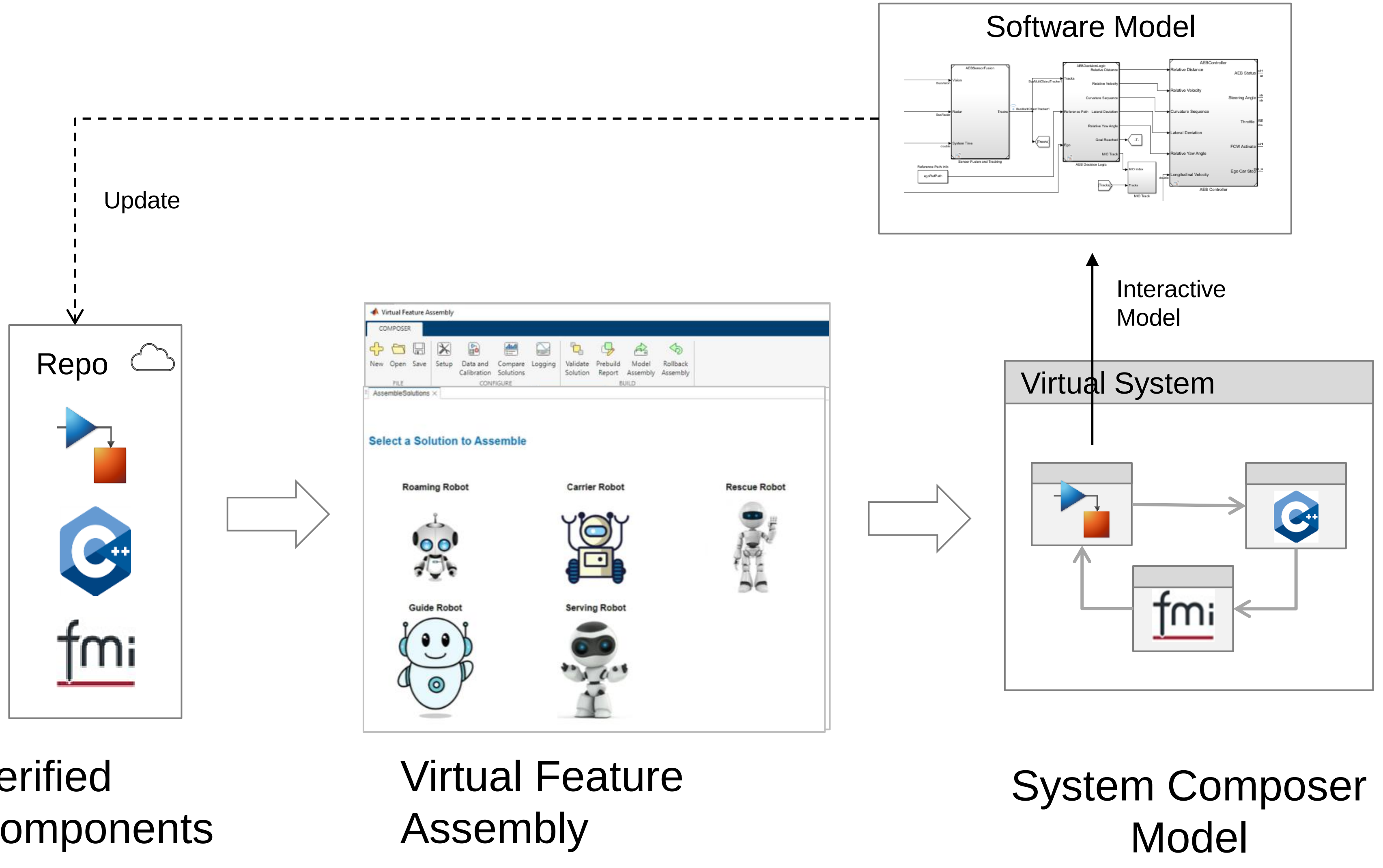
Select Vehicle features & characteristics

Your Component Under Test



Assembled Vehicle Model

Towards a fully distributed “model and code” integration framework



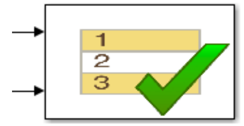
Verified Components

Virtual Feature Assembly

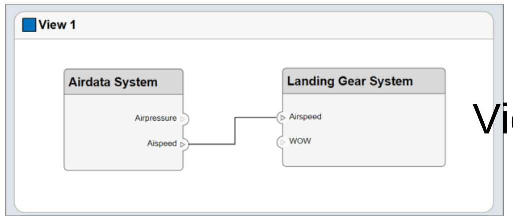
System Composer Model

Develop software functionality in context of virtual system

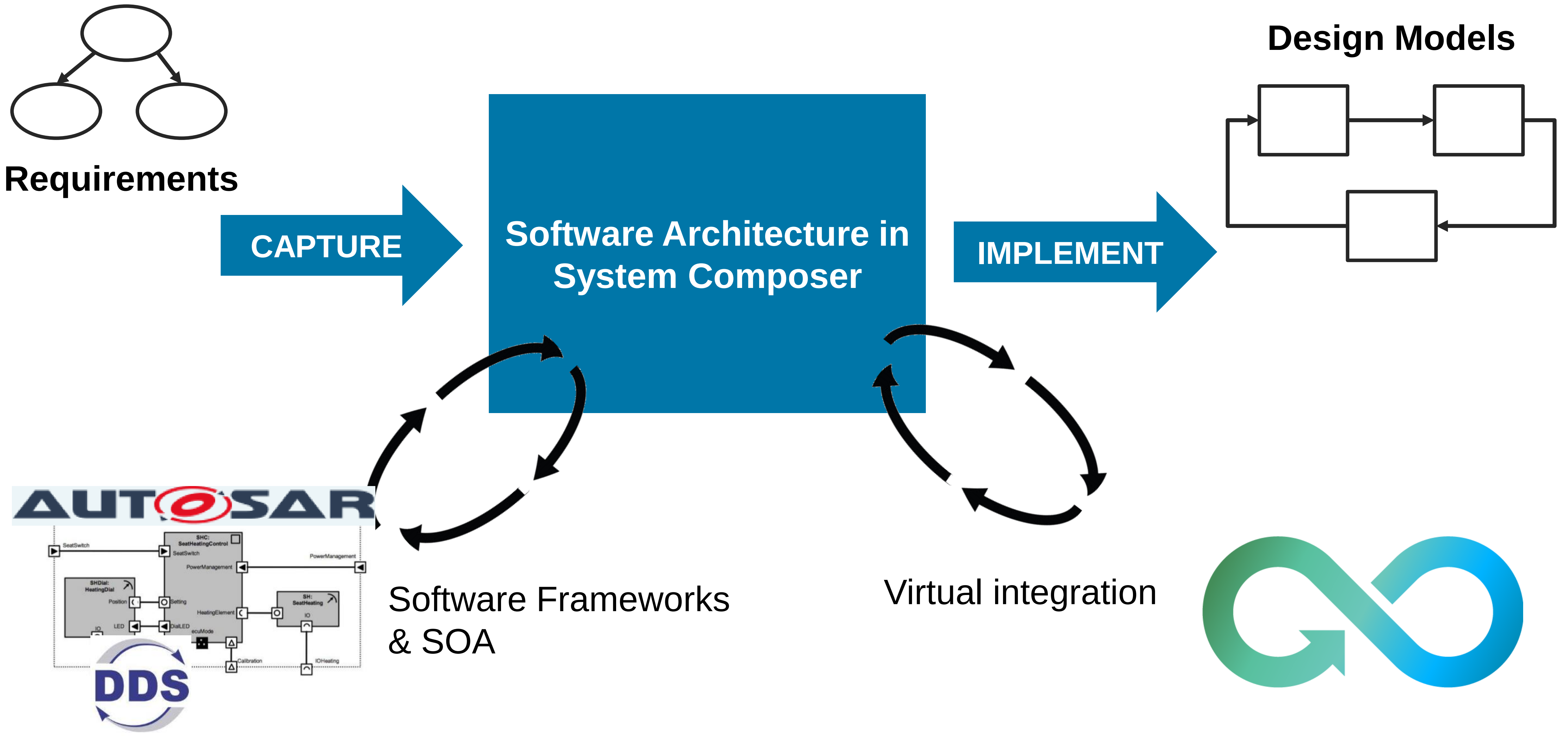
Simulation



Tests



Views


Requirements
CAPTURE
Software Architecture in System Composer
IMPLEMENT
Design Models
AUTOSAR
SHC: SeatHeatingControl
SeatSwitch
SeatSwitch
PowerManagement
PowerManagement
SHDial: HeatingDial
Position
Setting
LED
DialLED
IO
HeatingElement
SH: SeatHeating
IO
ecuMode
Calibration
IOHeating
DDS
Software Frameworks & SOA
Virtual integration

How are we taking this journey?

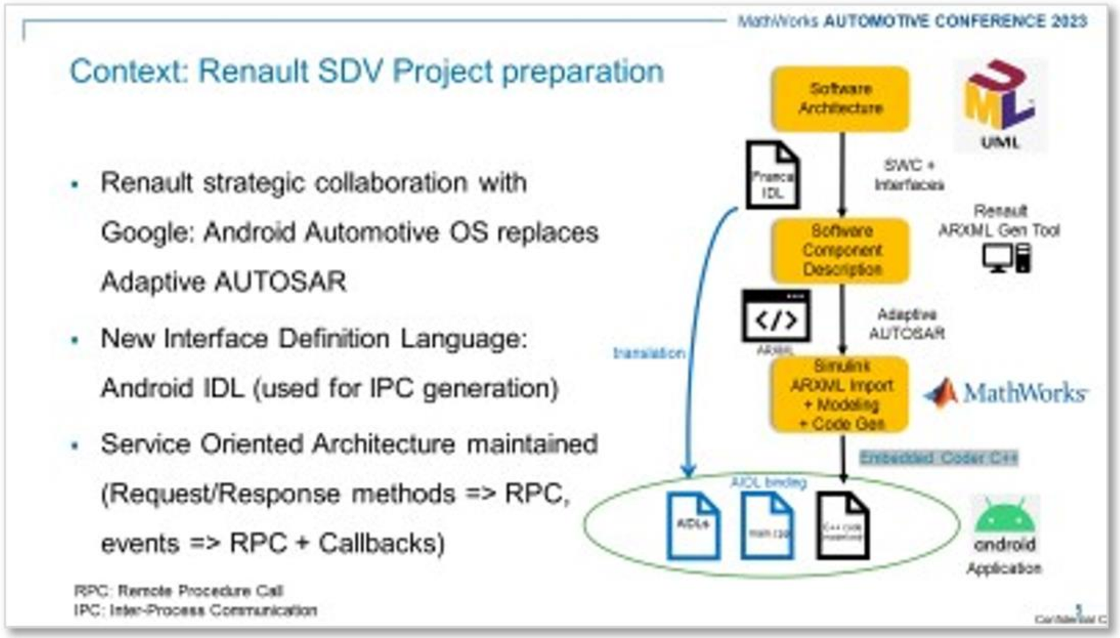
SDV: Integrating Simulink C++ Code in Android Automotive Environment

Rémy Brugnon, Renault Group

ZEEKR

Use Model-Based Design to Develop SOA Application Running on In-vehicle OS

Yining Luo, ZEEKR TECHNOLOGY LIMITED

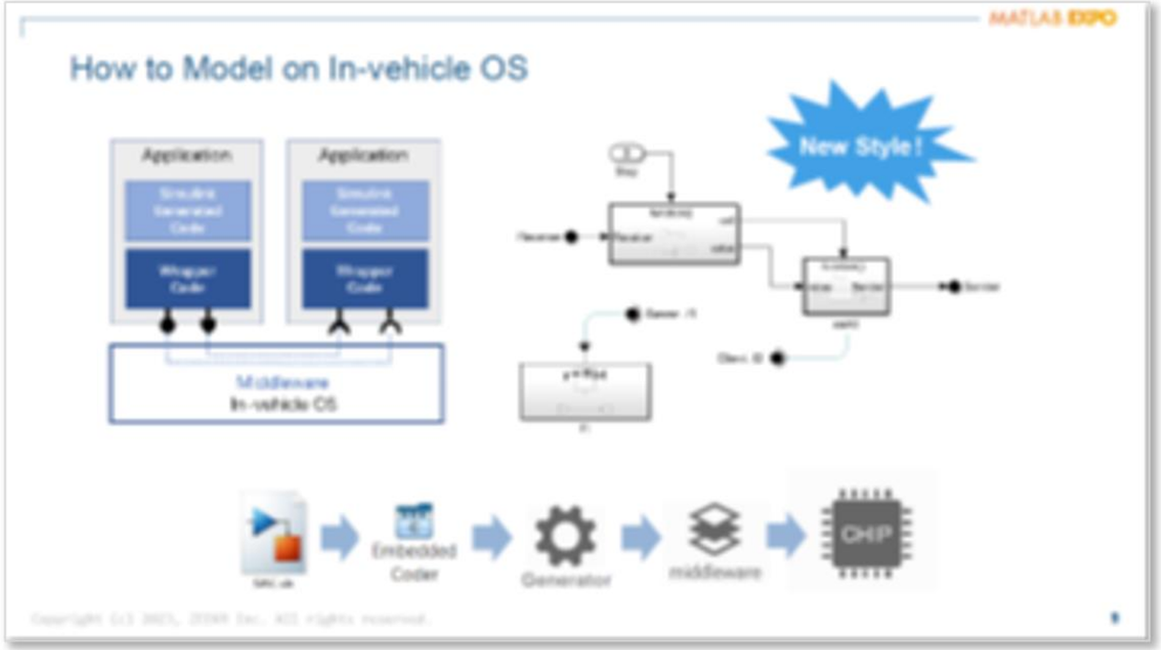


MathWorks AUTOMOTIVE CONFERENCE 2023

Context: Renault SDV Project preparation

- Renault strategic collaboration with Google: Android Automotive OS replaces Adaptive AUTOSAR
- New Interface Definition Language: Android IDL (used for IPC generation)
- Service Oriented Architecture maintained (Request/Response methods => RPC, events => RPC + Callbacks)

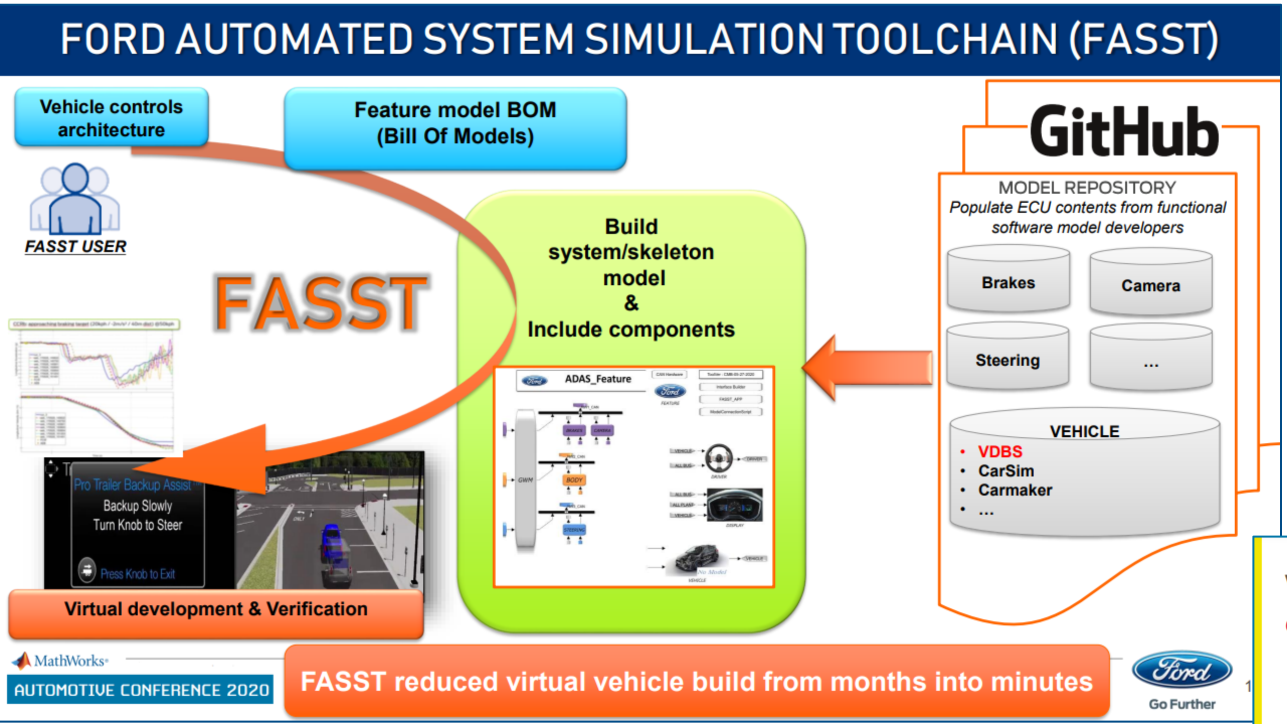
RPC: Remote Procedure Call
IPC: Inter-Process Communication



MATLAB EXPO

How to Model on In-vehicle OS

Industry practice examples



FORD AUTOMATED SYSTEM SIMULATION TOOLCHAIN (FASST)

Vehicle controls architecture

Feature model BOM (Bill Of Models)

FASST USER

FASST

Build system/skeleton model & Include components

ADAS_Feature

GitHub

MODEL REPOSITORY

Populate ECU contents from functional software model developers

Brakes

Camera

Steering

...

VEHICLE

- VDBS
- CarSim
- Carmaker
- ...

Pro Trailer Backup Assist

Backup Slowly

Turn Knob to Steer

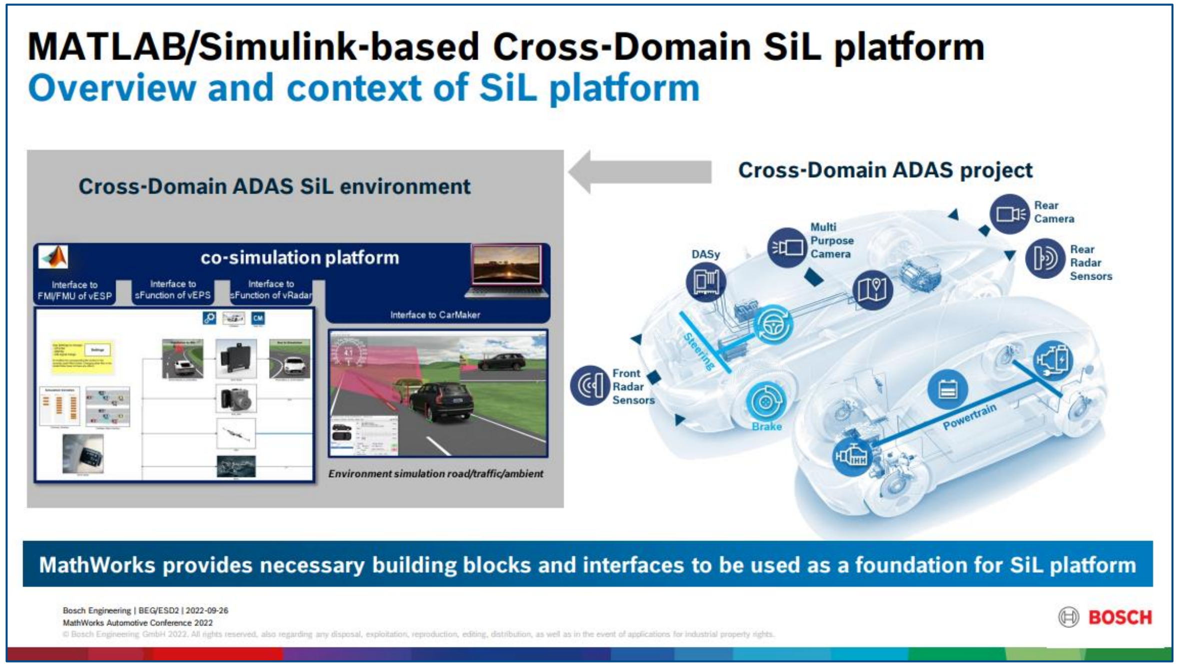
Press Knob to Exit

Virtual development & Verification

MathWorks AUTOMOTIVE CONFERENCE 2020

FASST reduced virtual vehicle build from months into minutes

Ford Go Further



MATLAB/Simulink-based Cross-Domain SiL platform

Overview and context of SiL platform

Cross-Domain ADAS SiL environment

co-simulation platform

Interface to FMI/FMU of vESP

Interface to sFunction of vEPS

Interface to sFunction of vRadar

Interface to CarMaker

Environment simulation road/traffic/ambient

Cross-Domain ADAS project

Rear Camera

Multi Purpose Camera

DASy

Rear Radar Sensors

Steering

Front Radar Sensors

Brake

Powertrain

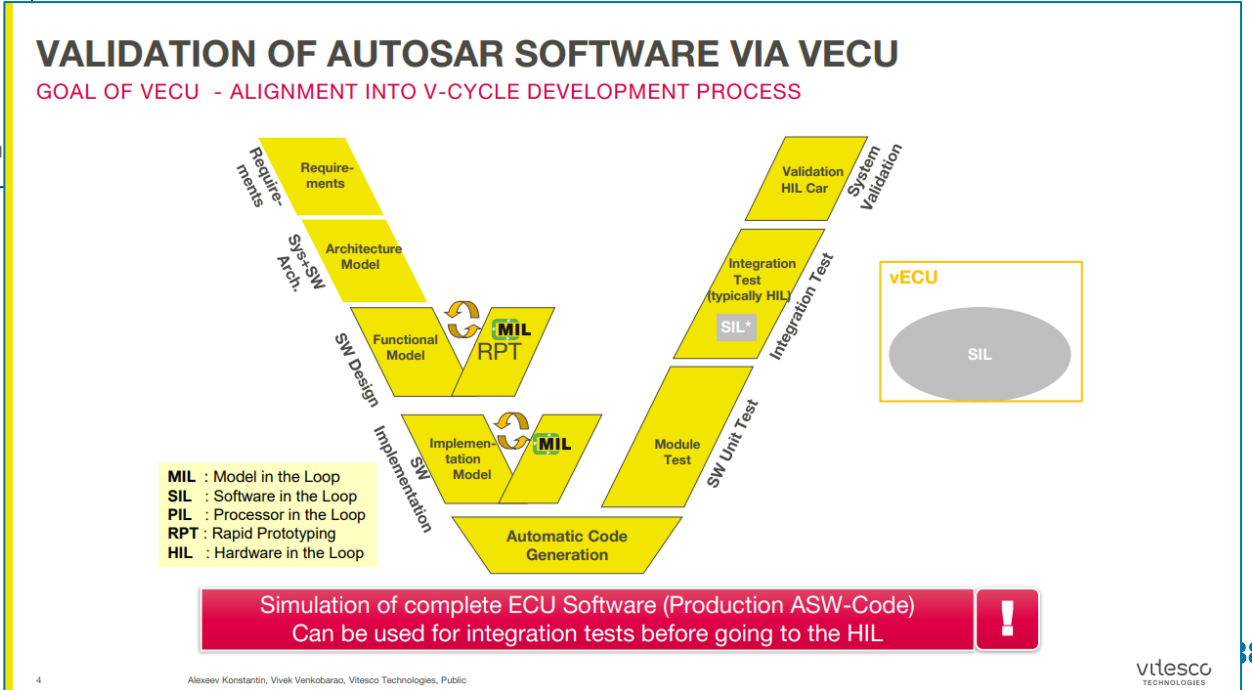
MathWorks provides necessary building blocks and interfaces to be used as a foundation for SiL platform

Bosch Engineering | BEG/ESD2 | 2022-09-26

MathWorks Automotive Conference 2022

© Bosch Engineering GmbH 2022. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH



VALIDATION OF AUTOSAR SOFTWARE VIA VECU

GOAL OF VECU - ALIGNMENT INTO V-CYCLE DEVELOPMENT PROCESS

Require-ments

Requirements

Sys+SW Arch.

Architecture Model

SW Design

Functional Model

MIL

RPT

SW Implementation

Implementation Model

MIL

Automatic Code Generation

SW Unit Test

Module Test

Integration Test

Integration Test (typically HIL)

SIL*

System Validation

Validation HIL Car

vECU

SIL

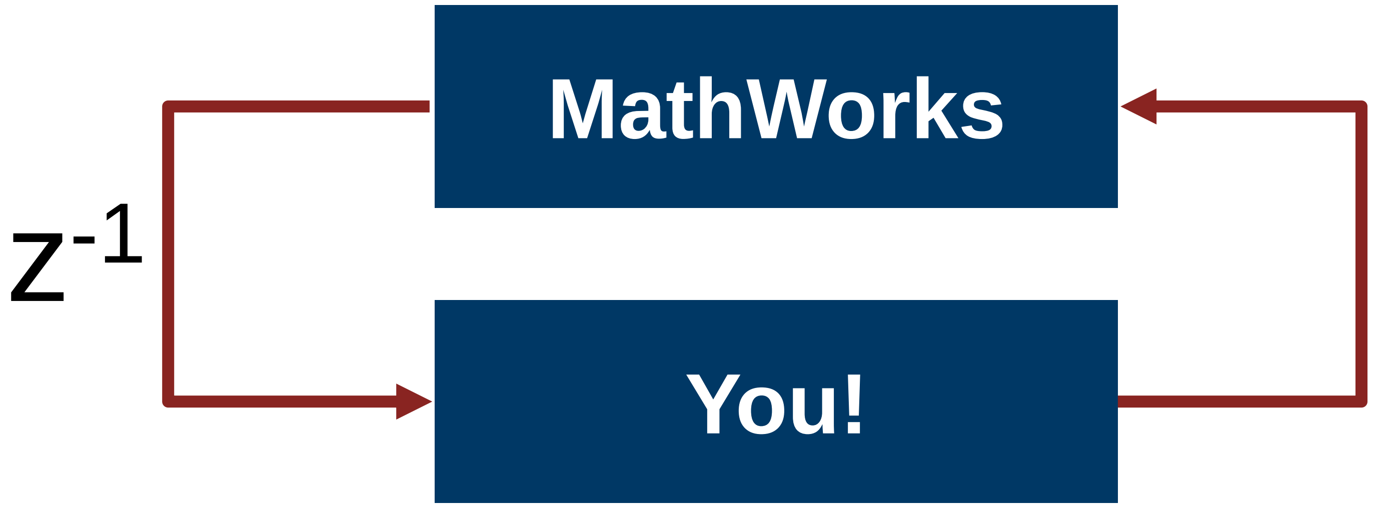
MIL	: Model in the Loop
SIL	: Software in the Loop
PIL	: Processor in the Loop
RPT	: Rapid Prototyping
HIL	: Hardware in the Loop

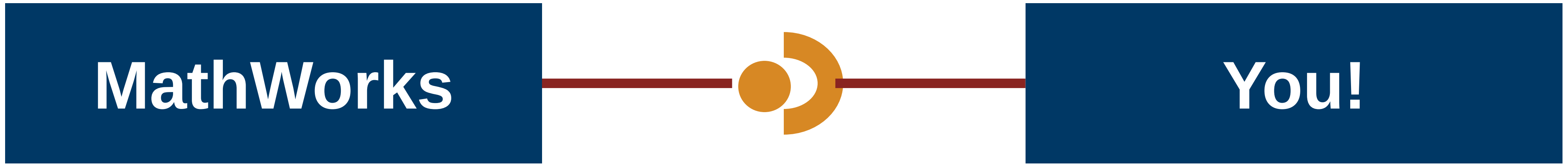
Simulation of complete ECU Software (Production ASW-Code)
Can be used for integration tests before going to the HIL

!

4 Alexeev Konstantin, Vivek Venkobarao, Vitesco Technologies, Public

vitesco TECHNOLOGIES


MathWorks
You!
z^{-1}


MathWorks
You!

Questions