ENOLA

MBSE Model Execution

February 2025

About US

LEAVE THE PAST IN THE PAST

IT'S TIME TO EVOLVE, DIGITALLY

WE ARE A TRAINING, COACHING, AND CONSULTING FIRM DEDICATED TO THE RAPID EVOLUTION OF OUR CLIENTS WITHIN THE DIGITAL UNIVERSE. OUR SERVICES ARE DIRECTED TOWARDS:

- DIGITAL ENGINEERING/TRANSFORMATION
- MODEL BASED SYSTEMS ENGINEERING
- ENTERPRISE ARCHITECTURE
- SOFTWARE ARCHITECTURE
- DATABASE ARCHITECTURE
- ONTOLOGIES
- COLLABORATION SERVER MANAGEMENT

MISSION

Enola WILL train, coach, and mentor your staff to be independently successful as quickly as possible.

Yes, our mission is to work ourselves out of a job!

Course Description

MBSE Model Execution is a four-day training designed to provide professionals with a background behind architectural simulation, and introduction to the Simulation Toolkit plugin, the simulation of multiple diagram types, co-simulation with MATLAB, and how to tie diagram simulations together to automate the architecture.

This course provides a mix of slides, instructor-led demonstrations, and hands-on labs. Our trainers are all experienced practitioners who understand the balance of theory and practicality.

Prerequisites:

Applying SysML with MagicDraw OR Enterprise Architecture in the UAF

Required Software:

No Magic's MagicDraw (version 19.0+) with the SysML plugin or equivalent No Magic or Dassault Systèmes CATIA Magic products.

Take-Aways:

- understanding the Simulation Toolkit plugin
- Working knowledge of architectural simulation and automation

AGENDA

DAY 1:

- Course Introduction
- Simulation of an Architecture
- Activity Diagrams
- State Machine Diagrams

DAY 2:

- Sequence Diagrams
- Parametric Diagrams
- Automated Requirement Verification via Simulation

DAY 3:

- Utilizing MATLAB and Simulink Co-Simulation
- Simulation Configuration Diagram

DAY 4:

- User Interface Modeling
- Action Language Helper

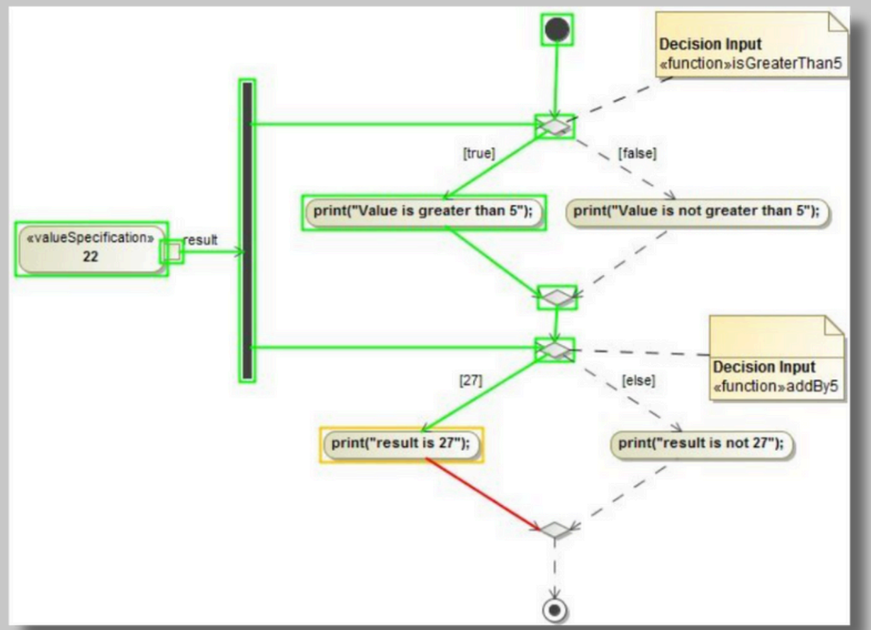
COURSE CONTENT

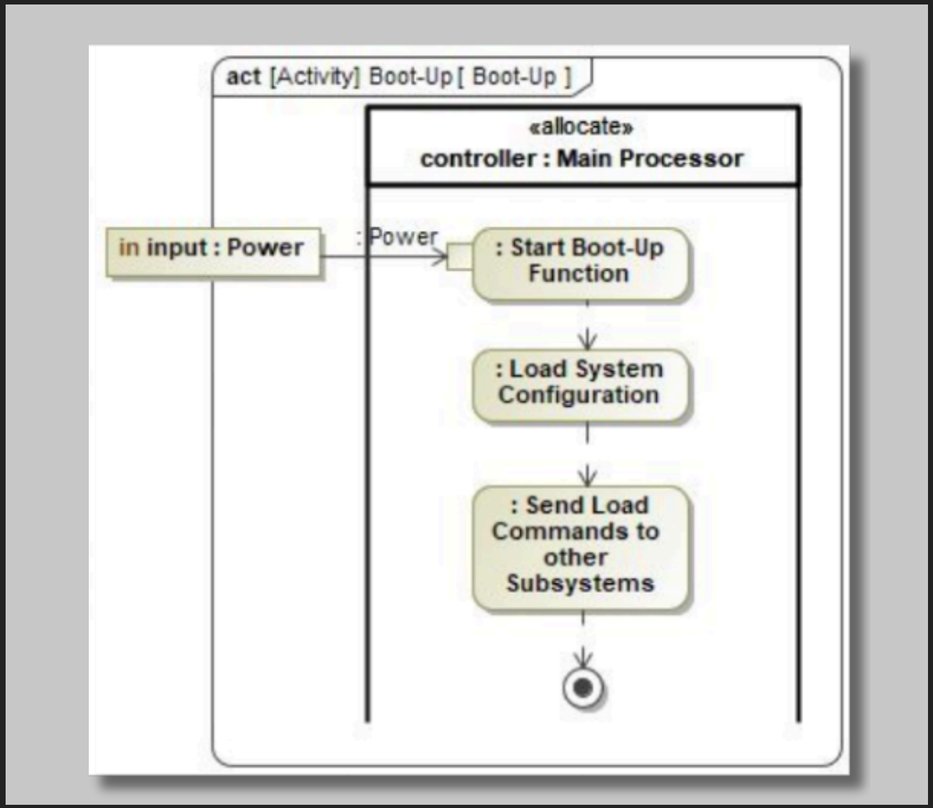
Simulation of an Architecture

Simulating the architecture enables users to verify that the architecture will meet the specified requirements and enables both understanding and communication with stakeholders.

This module covers:

- Purpose of Simulation
- Cameo Simulation Toolkit Key Features
- Simulation Sample Projects
- Simulation Project Template
- Model Simulation Engines





ACTIVITY DIAGRAMS

Activity Diagrams allow the definition of complex functional flow, and the simulation of these flows is essential to verify complex control logic.

This module covers:

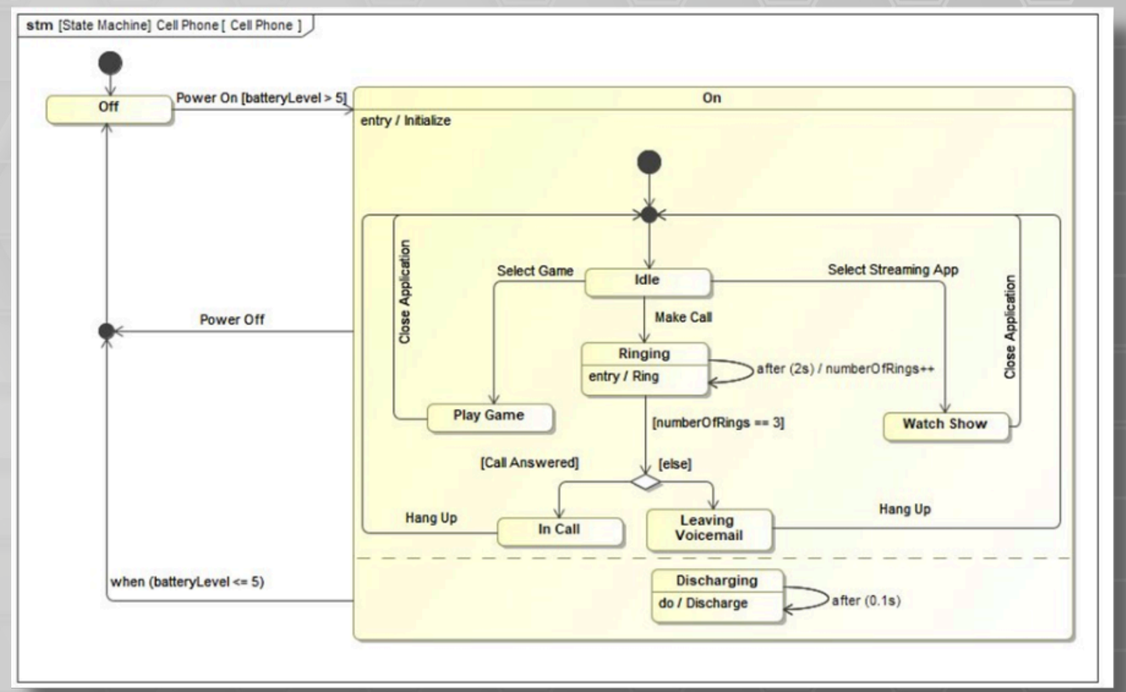
- Signals
- Operations / Signal Receptions
- Activity Diagram Review
- fUML
- Supported vs Unsupported Elements for Simulation

State Machine Diagrams

State Machines are behavioral elements that enable state-transition based behavior, making them excellent for driving the functionality of a system.

This module covers:

- State Machine Review
- Supported vs Unsupported Elements for Simulation

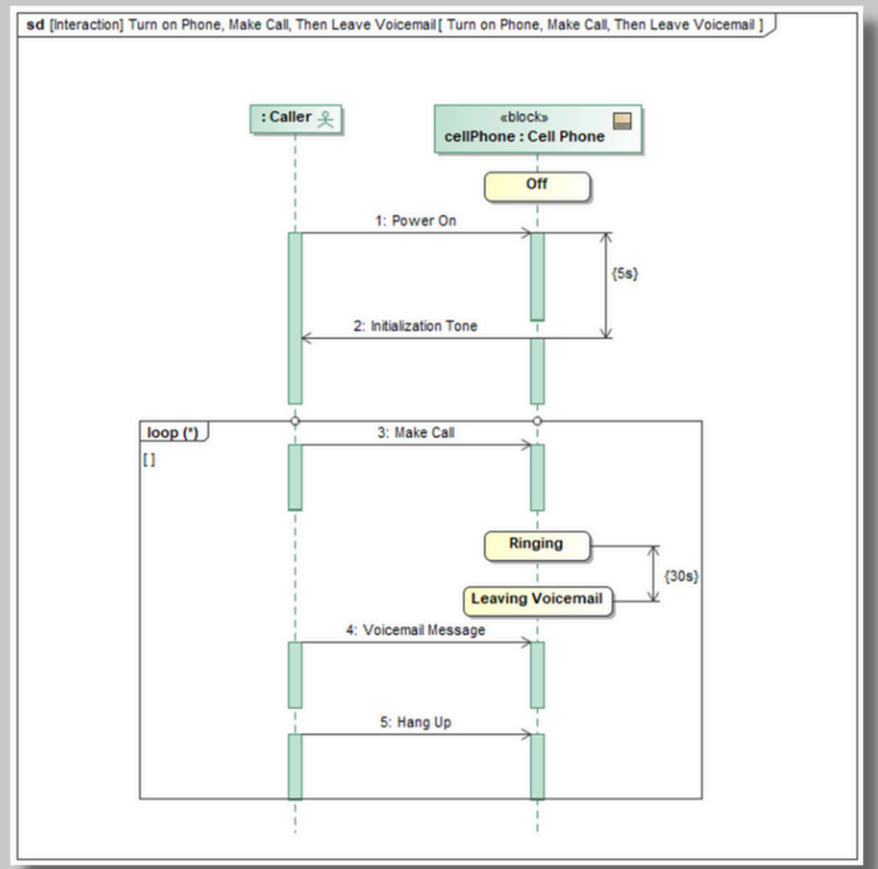


Sequence DIAGRAMS

Sequence Diagrams define instance-based message flows for behaviors with a strict sequence, like test case scenarios.

This module covers:

- Sequence Diagram Review
- Element Functionalities in Simulation
- Supported vs Unsupported Elements for Simulation

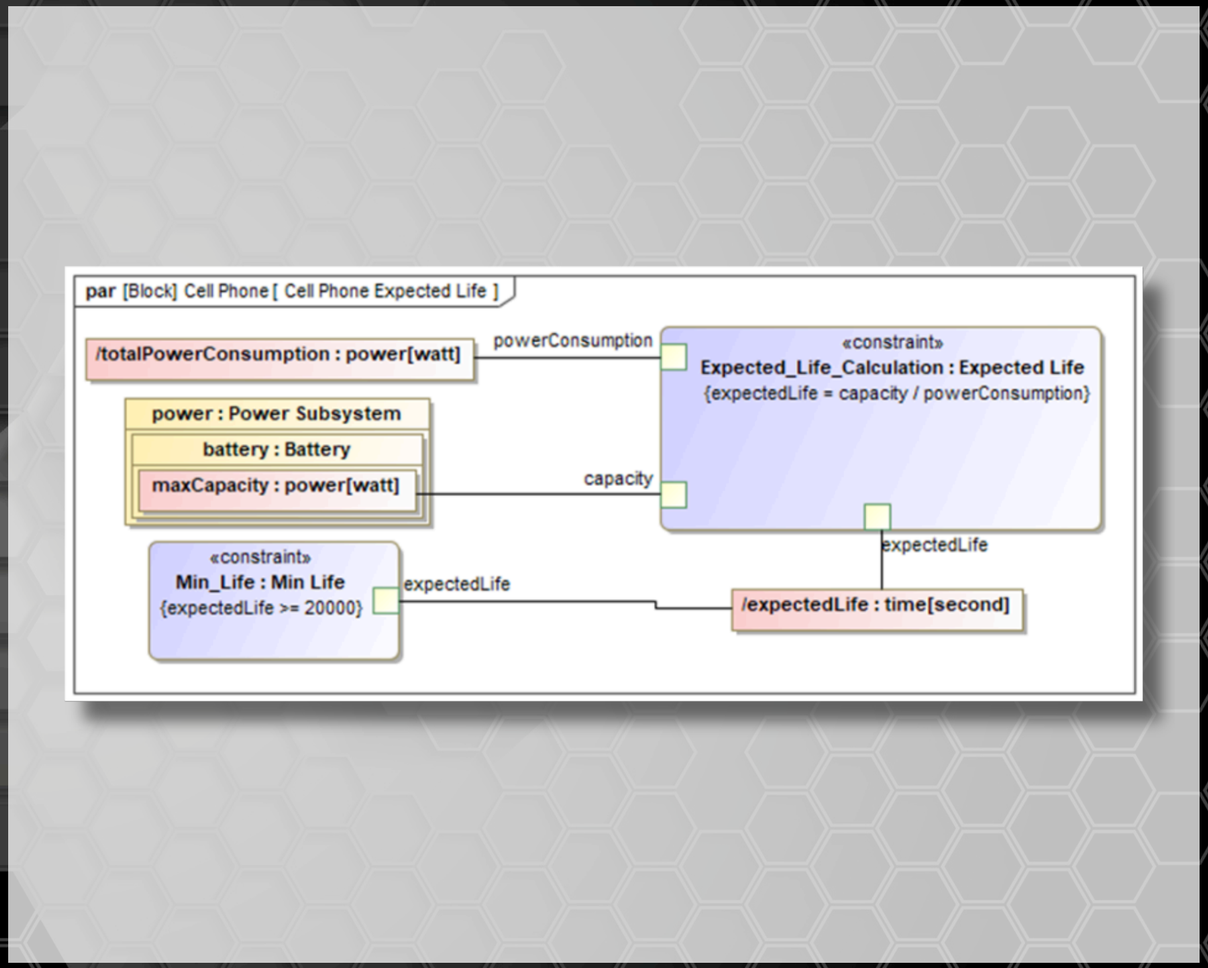


Parametric Diagrams

Parametric Diagrams are a special kind of internal block diagram used to bind value properties to constraint parameters to calculate quantitative system characteristics.

This module covers:

- Value Types / Enumerations
- Value Properties
- Constraint Blocks
- Constraint Properties
- Parametric Diagram Review



AUTOMATED REQUIREMENT VERIFICATION

Criteria

Scope (optional): REGULAR Requirements,COMMON_Requirements ... Filter: Context (optional): SUV_REGULAR ...

Requirement Verification: Pass Fail

#	Name	Text	Property	Bounds	Value	Margin
1	1 SUV_REGULAR Requirements					
2	1.1 Spring Coils	Spring shall have less than 8 coils.	suspension.spring.coils : Real	<8	7	1
3	1.2 Spring Deflection Distance	Spring shall have not more than 108-mm deflection distance.	suspension.spring.deflectionDistance : diameter[metre]	<=108	132	-24
4	1.3 Spring Free Length	The spring shall have a free length of 200 mm.	suspension.spring.freeLength : distance[millimetre]	=200	160	-40
5	1.4 Spring Outer Diameter	The diameter shall be less than 105 mm and more than 95 mm.	suspension.spring.outerDiameter : diameter[millimetre]	(95;105)	85	-10
6	1.5 Shock Absorber Length	Overall shock absorber length shall be at maximum of 600.	suspension.shockAbsorber.length : distance[millimetre]	<=600	450	150
7	1.6 Shock Absorber Weight	Shock absorber shall weight not more than 4 kg.	suspension.shockAbsorber.weight : mass[kilogram]	<=4	3	1
8	1.7 Tire Diameter	The tires shall have 18-inch rolling diameter.	suspension.wheel.tire.diameter : Integer	=18	17	-1
9	1.8 Tire Height	The tire height shall be not less than 45 and not more than 60.	suspension.wheel.tire.height : distance[millimetre]	[45;60]	50	5
10	1.9 Tire Width	The tire width shall be between 205 and 270 millimeters.	suspension.wheel.tire.width : distance[millimetre]	[205;270]	185	-20
11	1.10 Rotor Diameter	The brake rotors shall not exceed 0.28 meter diameter.	brake.rotor.rotorOuterDiameter : diameter[millimetre]	<=0.28	0.29	-0.01
12	1.11 Pad Center Length	The Pad Center Length shall be between 0.075 and 0.14 meters.	brake.pad.padLength : length[metre]	[0.075;0.14]	0.15	-0.01
13	1.12 Brake Pad Life	Brake pads shall have a projected life of at least 57500 km.	brake.pad.padLifeSpan : distance[kilometre]	>=57500	90000	32500
14	1.13 Pad Width	The Pad width shall be more than or equals 45e-3 and less than 65e-3 meters.	brake.pad.padWidth : diameter[metre]	(0.045;0.065)	0.042	-0.003

By combining text-based requirements with descriptive and analytical models in SysML, requirement verification can be automated.

This module covers:

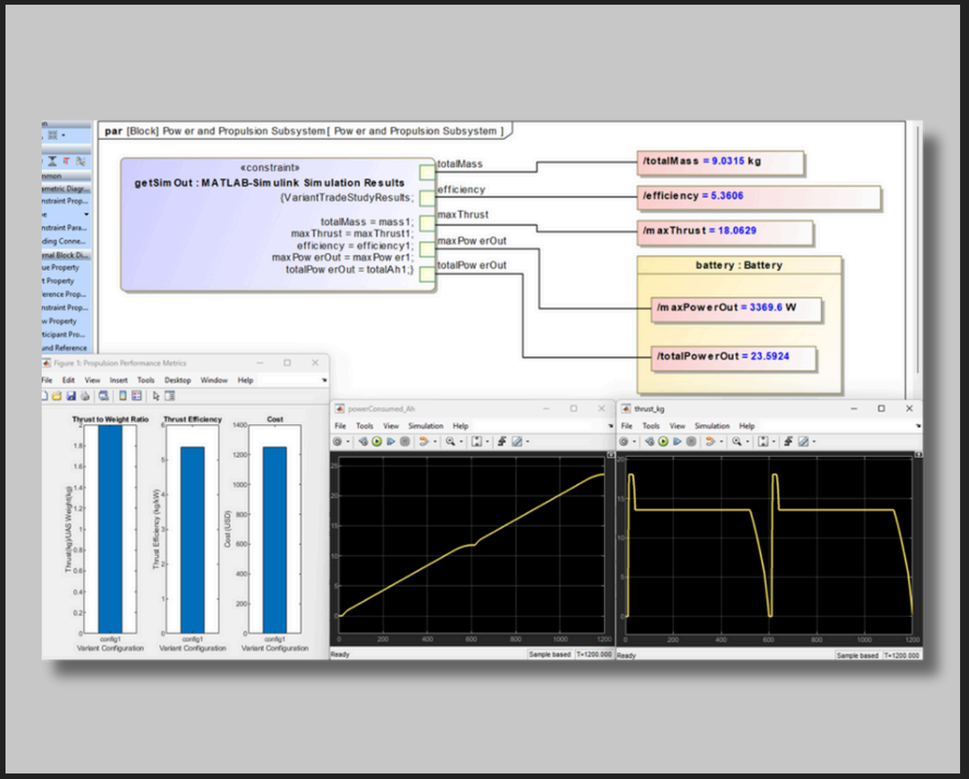
- Refining Requirements with Constraints
- Verification in Requirement Tables

Utilizing MATLAB and Simulink Co-Sim

For simulations requiring more advanced computation, headless co-simulation with MATLAB as the math engine can be launched to integrate MATLAB and Simulink models with the system architecture.

This module covers:

- MATLAB & Simulink Drag-n-Drop for Constraints & Actions
- Simulink Import for Internal Block Diagrams
- Shared MATLAB Sessions
- Utilizing GitHub with MATLAB & MagicDraw

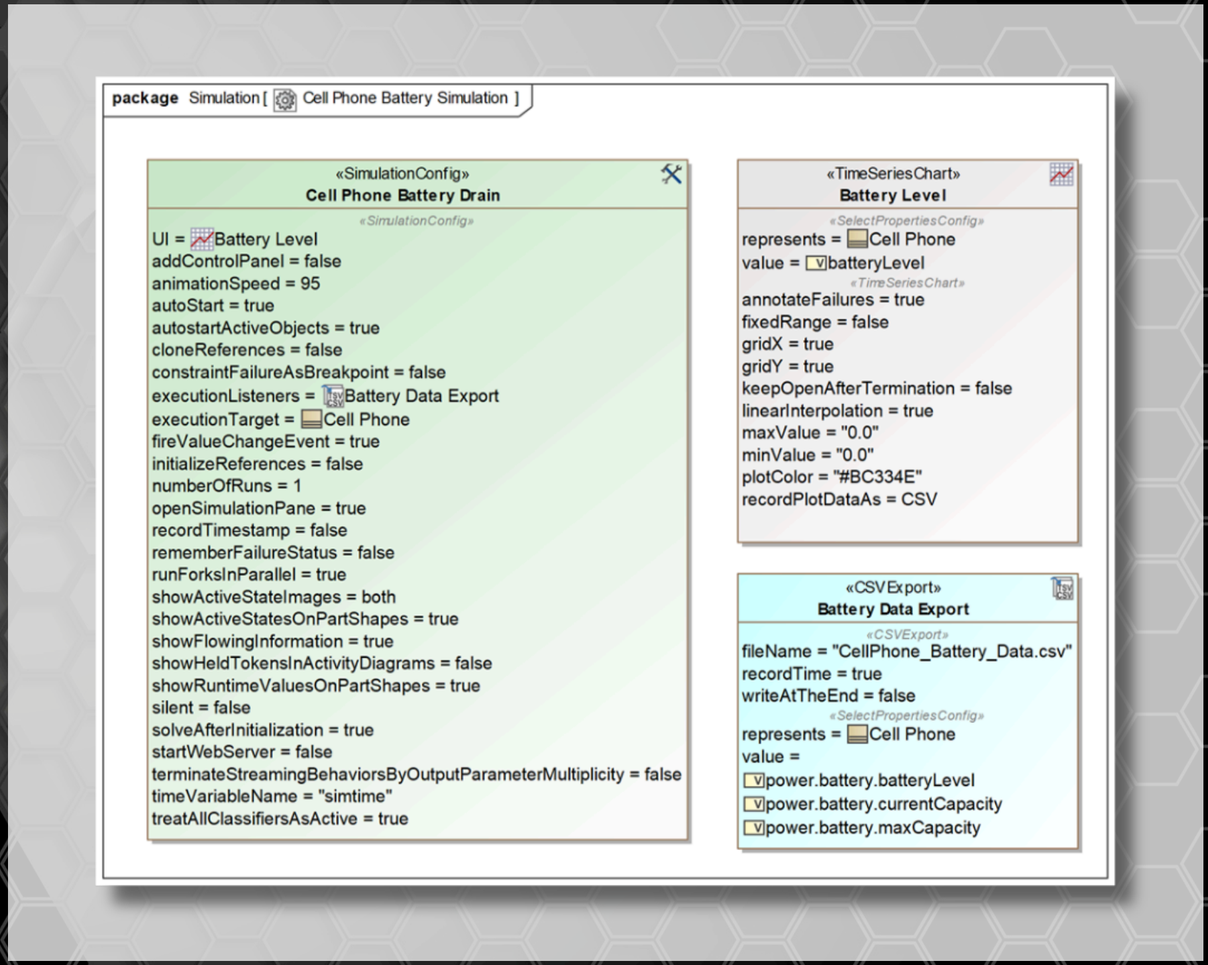


Simulation Configuration Diagram

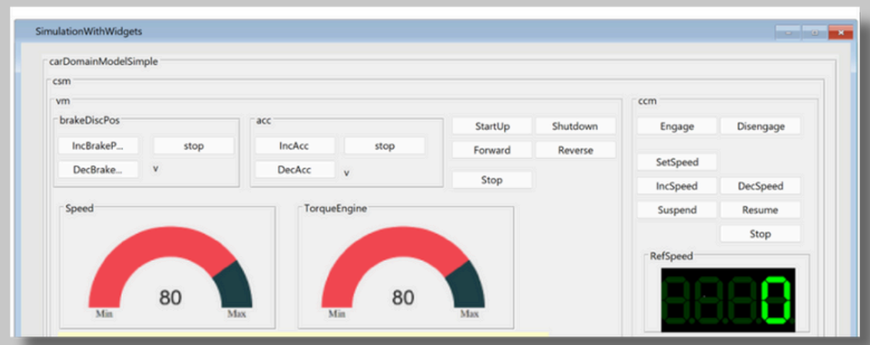
The Simulation Configuration Diagram allows for more advanced simulation with pre-configured options for execution and options for saving simulation data.

This module covers:

- Simulation Configurations
- Charts
- Image Switcher and Active Image
- Execution Listeners



User Interface Modeling Diagram



The User Interface Modeling Diagram allows for the design of Graphical User Interfaces to be shown during run time. These can display run-time values and control aspects of the simulation.

This module covers:

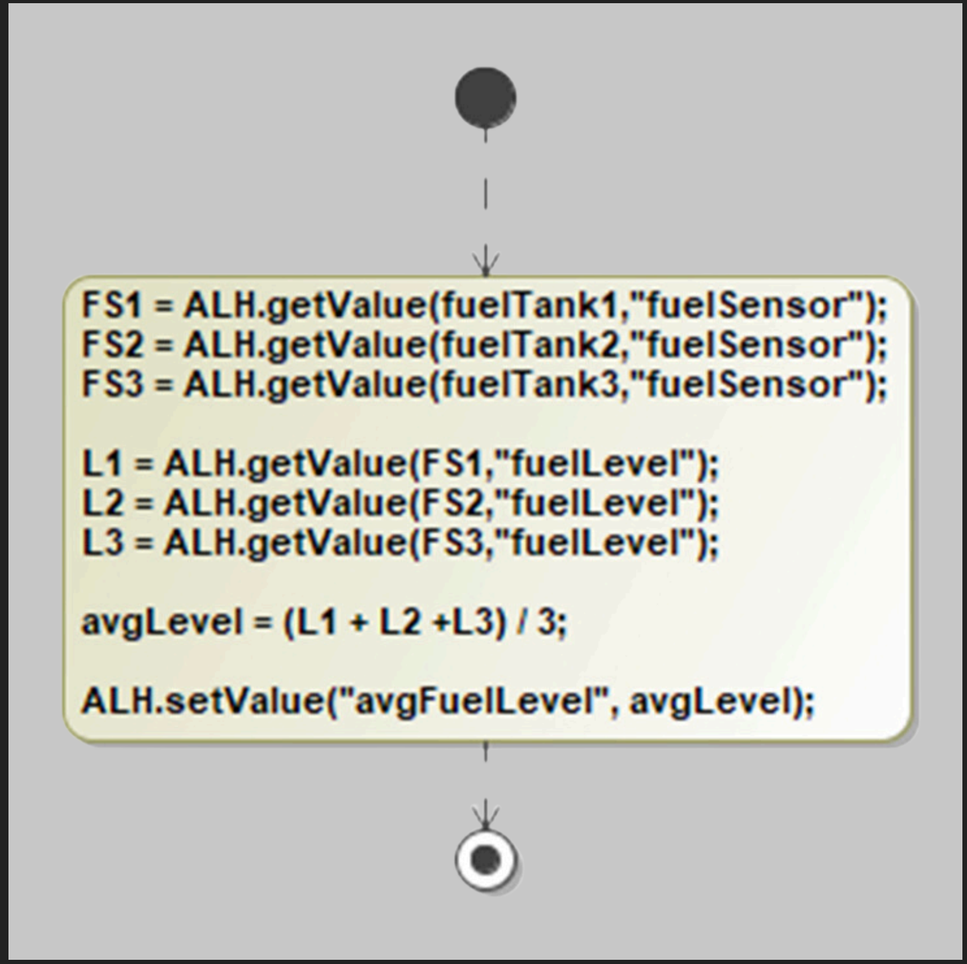
- Simulation Support
- Containers, Buttons, and Text
- Other UI Elements
- UI Control Hierarchy
- Structure to UI Hierarchy Example

Action Language Helper

The Action Language Helper (ALH) is a special API for model execution that allows for more complicated simulations and executable models.

This module covers:

- Methods
  - Get/Set Value
  - Get Tag Value
  - Create/Send Signal
  - Using Global Variables
- Predefined Variables
- Language Settings
- Unboxing in JavaScript Rhino



ENOLA

CONTACT US

www.enola.com

training@enolatech.com

+1 877 281 7341

linkedin.com/company/enolatech