ENOLA

Applying SysML with MagicDraw

May 2024

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

SysML with CATIA Magic is a three day course dedicated to training staff in the SysML language leveraging the CATIA Magic brand of tools (aka. MagicDraw/Cameo).

This course provides a mix of slides, instructor led demonstrations, and a lab in which students architect an example system. Our trainers are all experienced practitioners who understand the delicate balance of theory and practicality.

Prerequisites:

Foundation in basic systems engineering

Required Software:

CATIA Magic/No Magic modeling tool with SysML plugin

Take-Aways:

- Ability to understand and model the 9 SysML Diagrams and 3 traceability views
- Exposure to Simulation and Queries with CATIA Magic

AGENDA

DAY 1:
OVERVIEW & REQUIREMENTS

- Training Overview
- Introduction to MBSE and SysML
- MagicDraw/Cameo Walkthrough
- Model Organization with Packages
- Requirements

DAY 2:
STRUCTURE

- Use Case Functionality
- Structural Decomposition and Taxonomies
- Interface Definitions
- Constraints and Parametric Evaluation

DAY 3:
BEHAVIOR

- System Functionality Flow
- Interactions of Messages
- State Machines
- Querying and Analysis

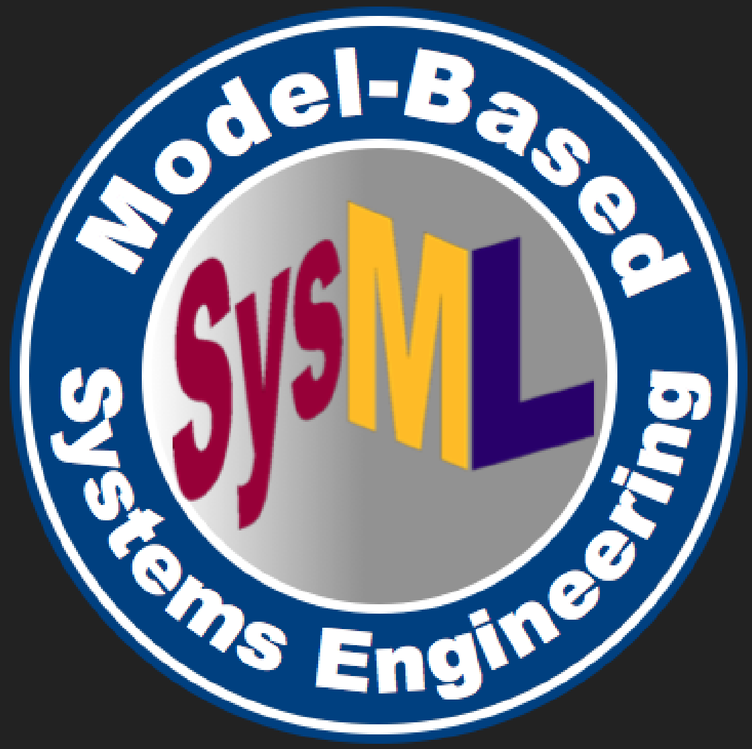
COURSE CONTENT

Introduction to MBSE with SysML

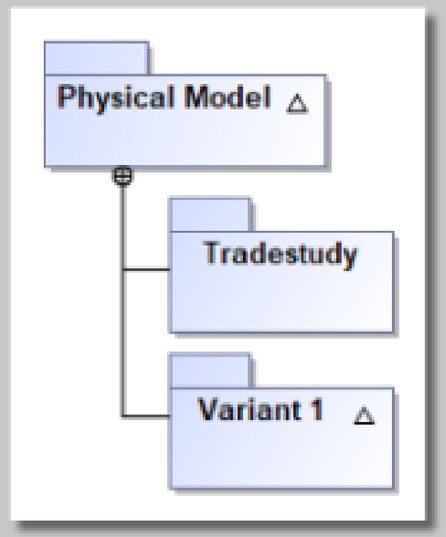
MBSE requires practitioners to select a language, a tool, and a methodology. This course focuses on the language (SysML) and tool (Catia Magic).

This module covers:

- MBSE
  - Definitions
  - Advantages
- Introduction to SysML



Model ORGANIZATION



Packages, Models, Libraries, and Profiles enable users to define the structure of the database to improve navigability and data grouping.

This module covers:

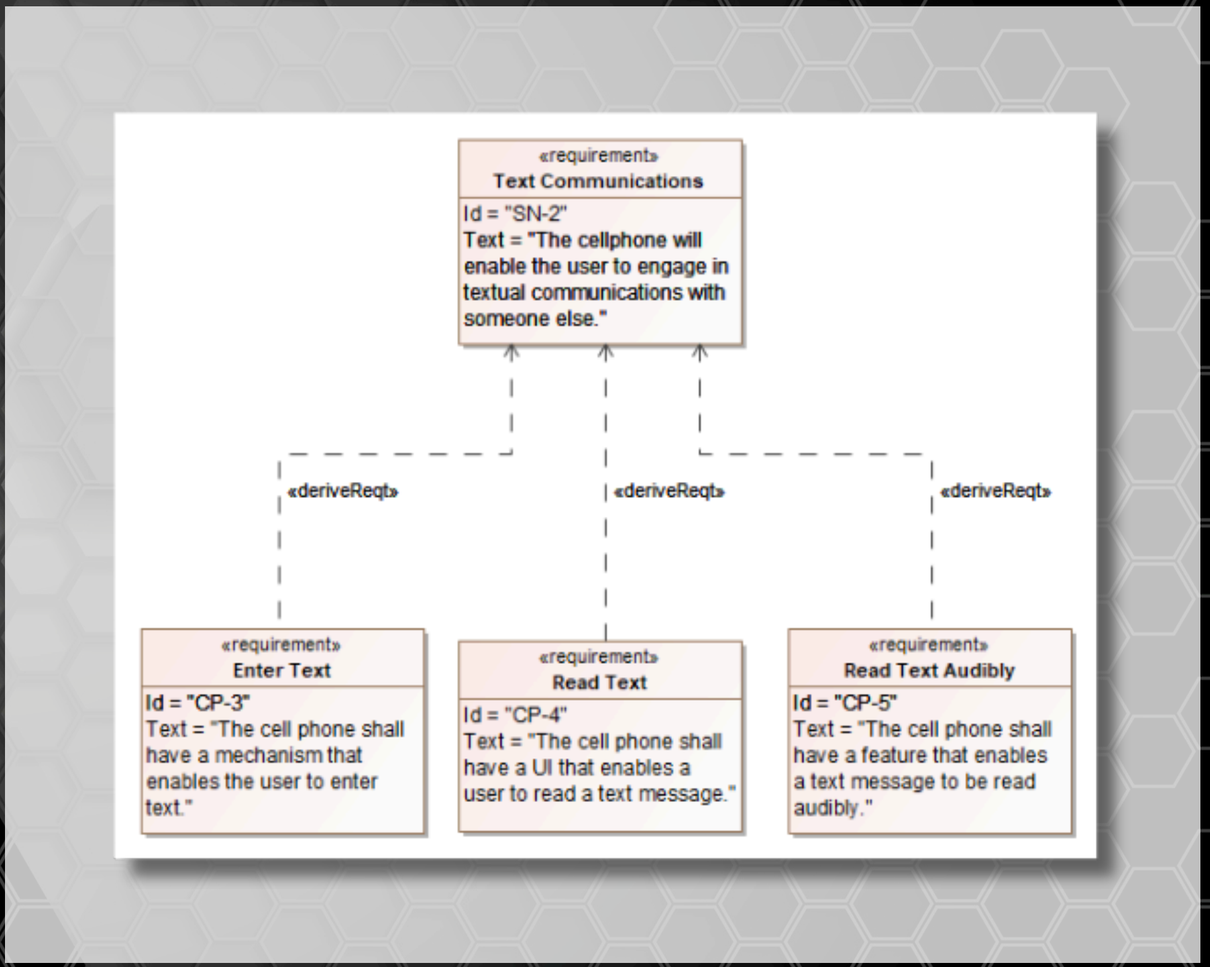
- Differences between the Package and it's customizations
- Definition of a Namespace and how that impacts the project
- Relationships between Packages Best practices for model organization

Requirements

Requirements are text-based specifications that constrain the system architecture.

This module covers:

- Requirements, their fields, and Extended Requirements
- Relationships between requirements and from model elements to requirements
- Requirement Customizations
- Alternate views (maps, tables, and matrices)

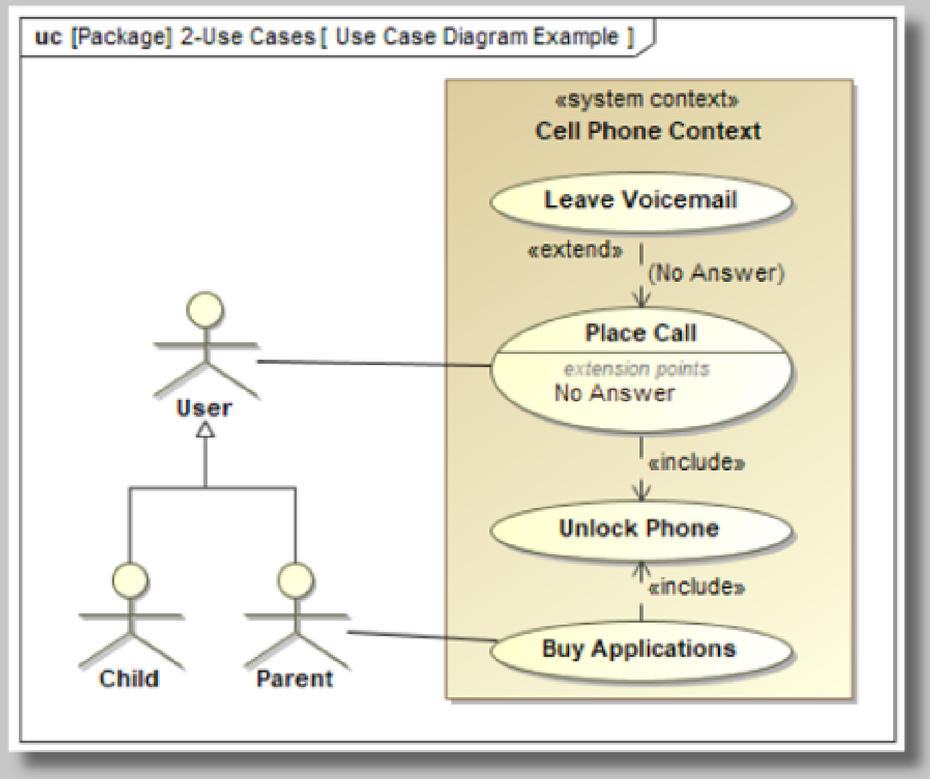


Use Cases

Use Cases are leveraged to capture interaction scenarios between external systems, users, and the system being architected.

This Module Covers:

- Use Cases
- Actors
- System Context
- Applicable Relationships

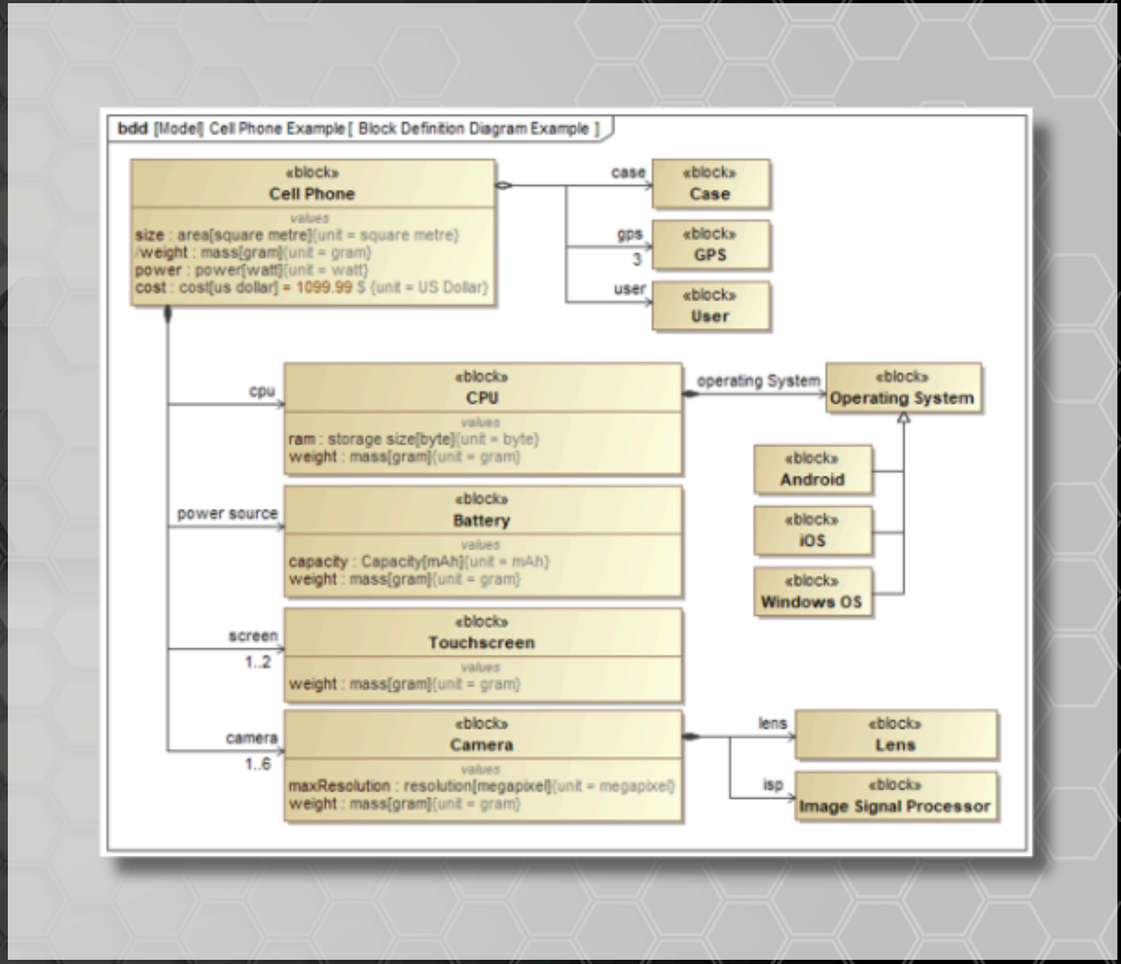


Structures

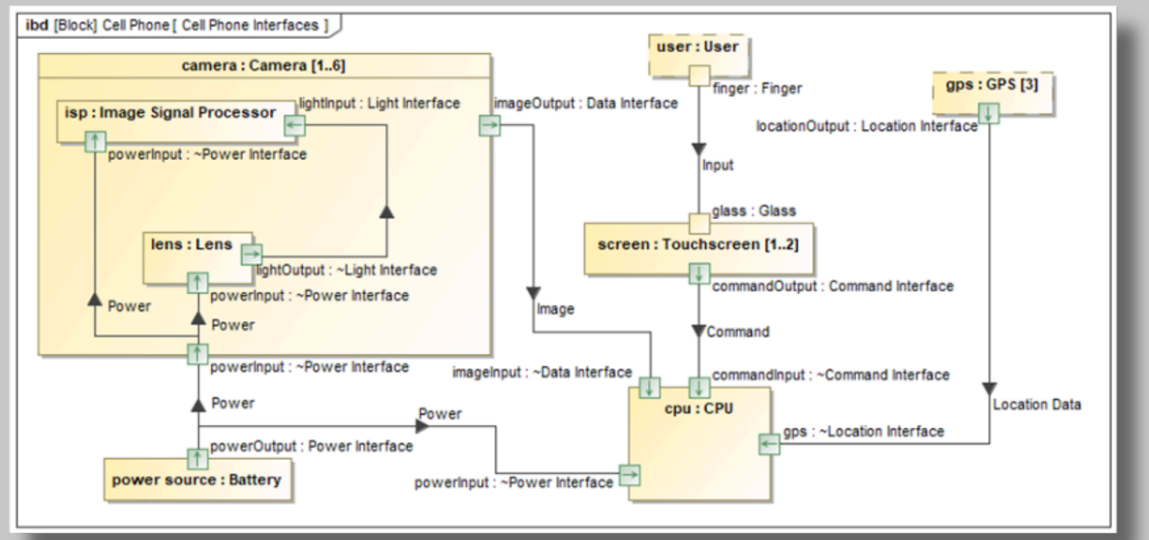
The structural pillar specifies the composition of the system through its subsystems and components.

Within this module we cover:

- Blocks and their features
- Associations and properties
- Instances



Interface DEFINITIONS



Interfaces are parts on the boundary of a structure that enable matter, energy, or data to pass to enable their functionality.

Within this module we cover:

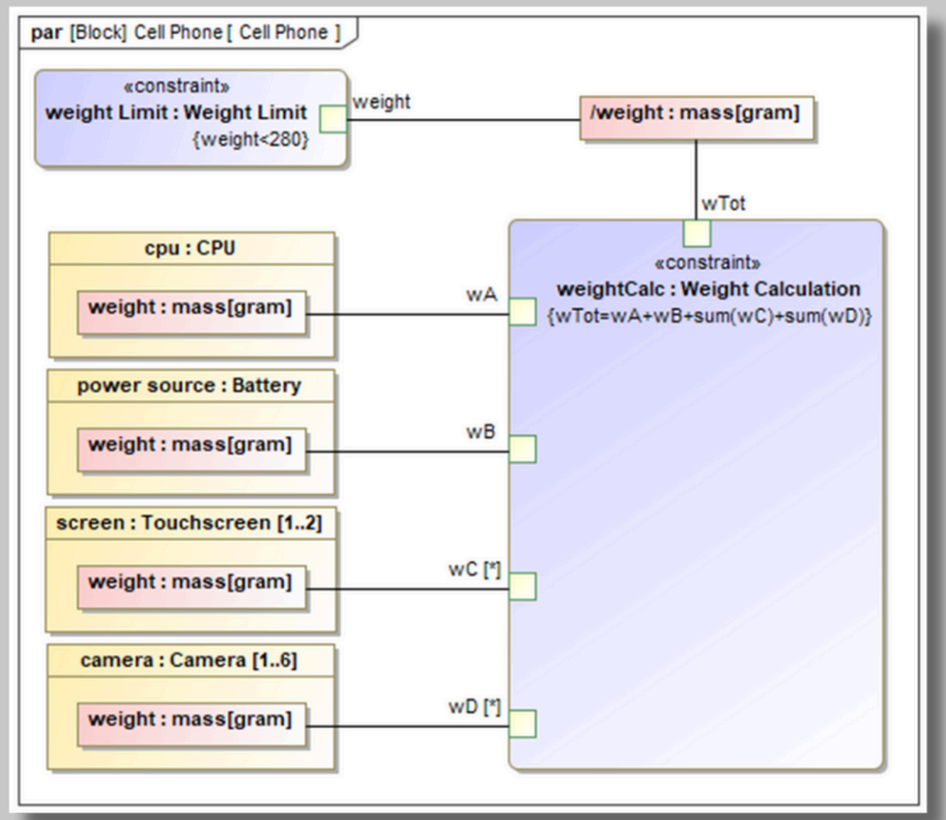
- Interface Definitions
- Internal Block Diagrams
- Connectors / Item Flows

Parametrics

Parametrics enable us to define and verify the mathematical model behind the system architecture.

Within this module we cover:

- Constraint Definitions
- Parametric Diagrams
- Verification of the performance and structural requirements

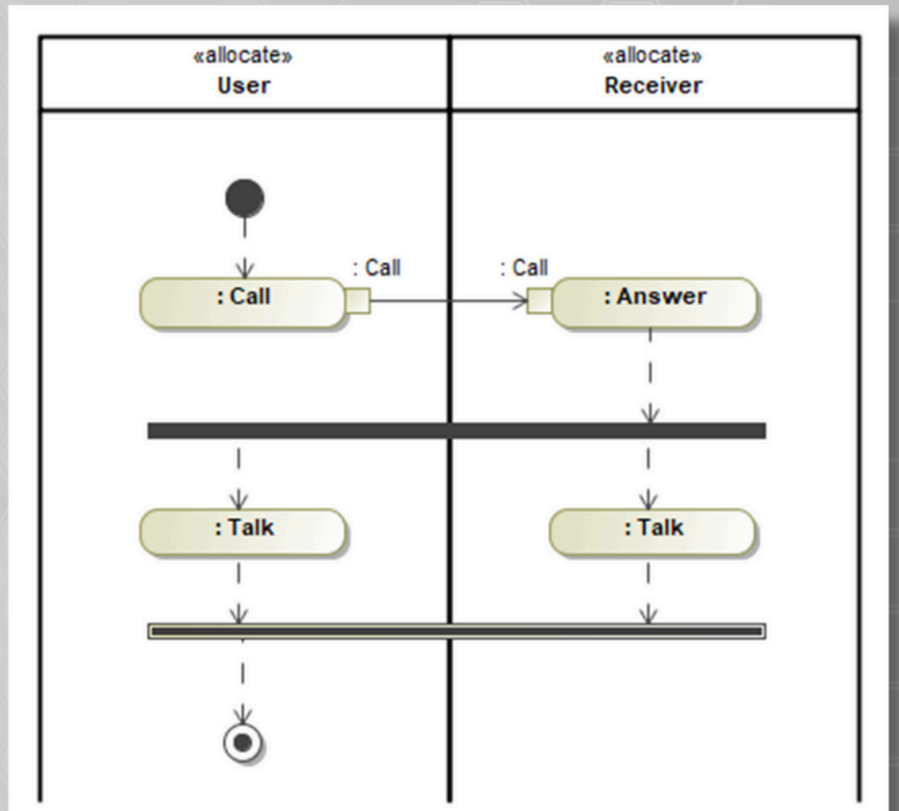


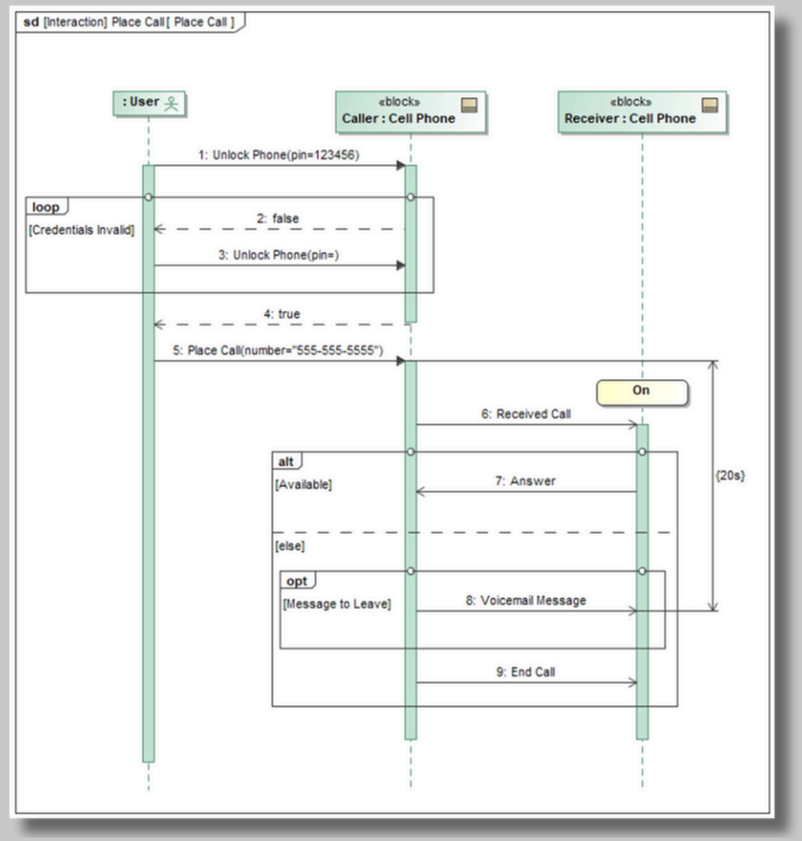
System Functionality FLOW

Activity diagrams enable us to specify the action a structure performs, the sequence of actions, and the information that is either produced or consumed by a given action.

Within this module we cover:

- Activity Diagrams and their Elements/Relationships
- Connections back to Use Case Diagrams
- Activity Allocation to Structures





Interactions of Messages

Instance-based message flows for test Test Case scenarios are quite important to verify functional requirements.

Within this module we cover:

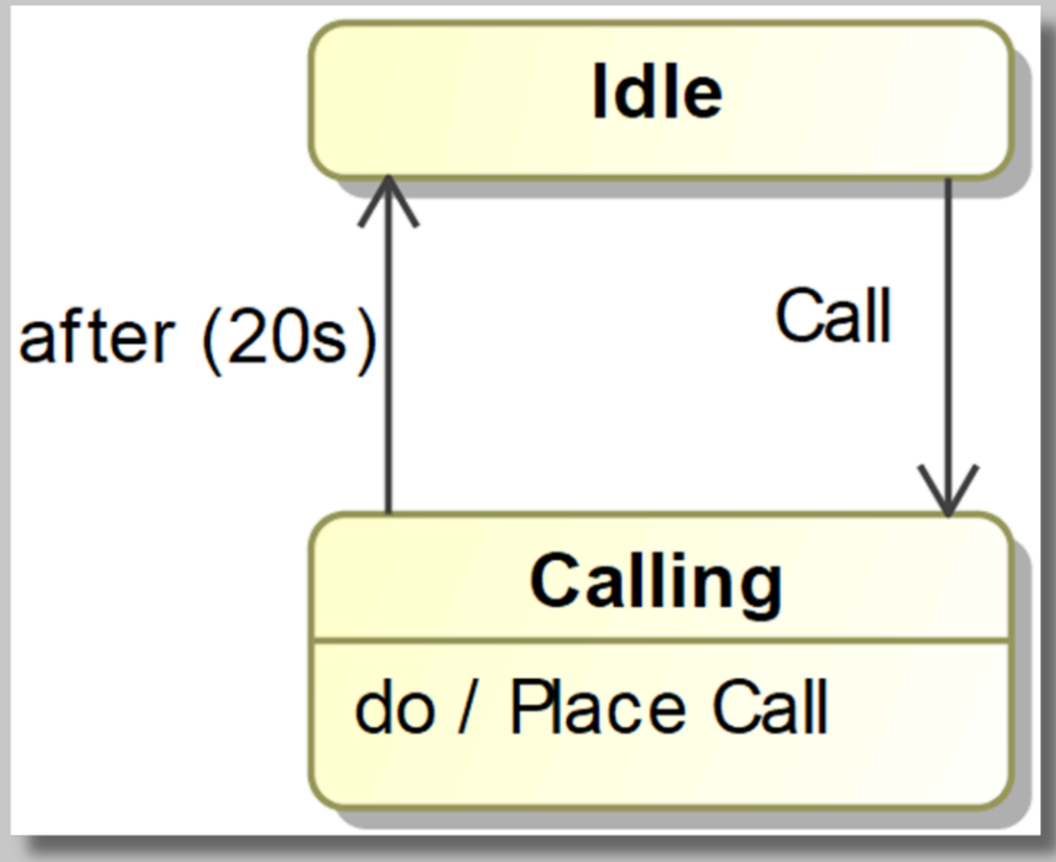
- Sequence Diagrams and their Elements/Relationships
- Reuse of modular interactions
- Constraints (state invariants / duration and time constraints)

State Machines

State Machines enable us to better understand the significant conditions of a structure, which behavior it can perform in these conditions, and what causes it to transition from one state to another.

Within this module we cover:

- State Machine Diagrams and their Elements
- Integration to other Behaviors

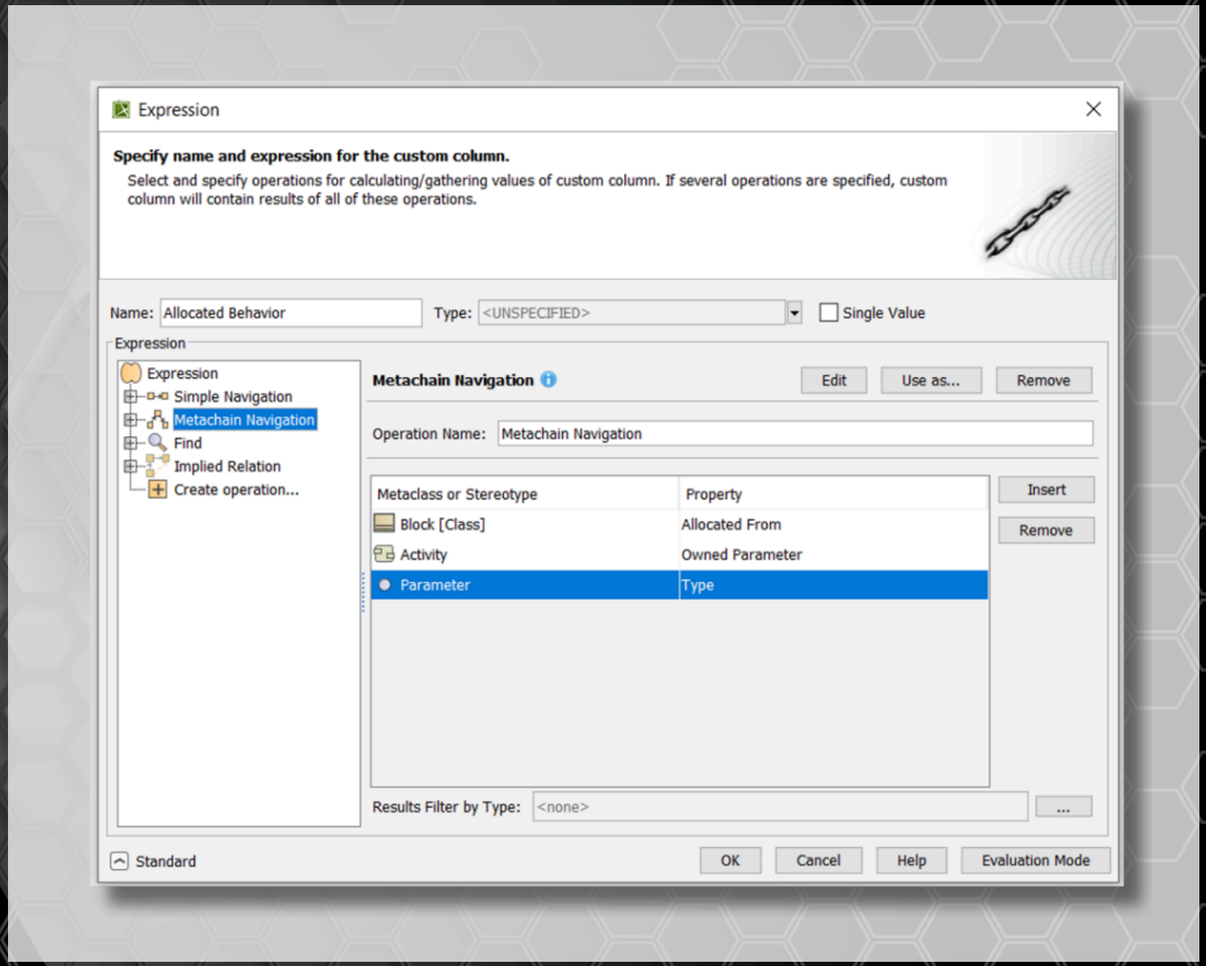


Querying & ANALYSIS

Model Queries enable us to create additional linkages between elements through their relationships, automate traceability views, and perform other script-based automations.

Within this module we cover:

- Querying
- Analysis



ENOLA

CONTACT US

www.enola.com

training@enolatech.com

+1 877 281 7341

linkedin.com/company/enolatech