

## SysML Training | Systems Modeling Language

**Length:** 3 Days

**Summary:** The Systems Modeling Language (SysML) is a modeling language useful for Systems Engineering applications supporting the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems (SoS). SysML for Systems Engineering includes Systems modeling, System Analysis and System Design as essential enabling techniques for systems engineering processes Systems Modeling Language (SysML) a subset of UML for Systems Engineering). SysML Training Course provides technical details of SysML a systems engineering modeling language. The syntax of SysML is covered and each concept is explained through a number of hands-on practical application workshops.

Learn about:

- UML and SysML
- Model-based systems engineering (MBSE) approach
- SysML diagrams and language
- Look at SysML from the systems engineering process viewpoint
- Develop a system conceptual model and architecture using SysML.
- System architecture, modeling and design with SysML

The SysML training course covers the following:

- Provides a comprehensive overview of the SysML concepts, terminology and modeling notation
- Shows attendees how to create analysis and design models with SysML
- Covers the complete system-modeling lifecycle from requirements to Validation of the system
- Presents many practical case studies

**Learning Objectives:** Upon completion of this course, the attendees will be able to:

- Describe UML
- Describe SysML
- Describe model-based systems engineering approach
- List SysML diagrams and language concepts
- Apply SysML as part of a model based SE process
- Explore transitioning to SysML
- Synthesize and analyze existing architecting approaches to enhancing creativity while reducing ambiguity and complexity.
- Utilize out-of-the-box holistic system thinking
- Develop a system conceptual model and architecture using SysML.
- Define system architecture, modeling, form, function, structure and behavior with SysML
- Describe how a system's function emerges from its form and behavior
- Describe the notions of system, product, service, and project with SysML
- Model a combined Project-Product Lifecycle Management system and study the benefits of the project-product synergies with SysML
- Work with real life projects using SysML

**Who Should Attend:** Systems engineers, system modelers, system architects, software engineers, system integrators, analysts and designers, product developers, project leaders, project heads,

---

## COURSE CONTENT

### Overview of Systems Engineering (SE)

- Systems Engineering Overview
- Model-Based Systems Engineering
- SysML Language Overview
- SE Practices for Describing Systems
- Specifications
- Interface requirements
- System design
- Analysis & Trade-off
- Test plans
- Stakeholders Involved in System Acquisition

### What is Model-based systems engineering (MBSE)?

- What is a model?
- principles behind MBSE
- Formalized application of modeling to support system requirements, design, analysis, verification and validation activities
- SE Artifacts and transitions to MBSE
- MBSE Across the
- System Life Cycle Specifications
- Interface requirements
- System design
- Analysis & Trade-off
- Test plans
- MBSE benefits
- Modeling at multiple levels of the System
- Operational model
- System model

- Component model
- MBSE to support complex predictive and affects-based modeling
- Relationship between SysML and MBE/MBSE

### Overview of UML and SysML

- Diagram Overview and Language Concepts
- What is SysML?
- UML for Systems Engineering R
- SysML Diagram Taxonomy
- A subset of UML with extensions
- SysML as a UML Profile
- Systems including hardware, software, data, personnel, procedures, and facilities
- SysML in specification, analysis, design, verification, and validation of systems

### Overview of System Modeling with SysML

- Functional/Behavioral Model
- Performance Model
- System model
- Structural/Component Model
- Other Engineering Analysis Model
- Model Based Systems Engineering Benefits
- Shared understanding of system requirements and design
- Assists in managing complex system development
- Improved design quality
- The Structure of an element or system
- System

- Hardware

- Software

- Data

- Procedure

- Facility

- Person

### **SysML Diagram Techniques**

- Use Case
- Requirement
- Activity
- Block Definition
- Internal Block
- Sequence
- State Machine
- Parametric
- Package
- Allocation Tables

### **SysML Modeling Elements**

- Allocations
- Rationales
- Diagram Frames
- Model Views and Viewpoints
- Problems

### **SysML Diagram Taxonomy**

- Behavioral Diagram
- Activity Diagram
- Sequence Diagram
- State Machine Diagram
- Use Case Diagram
- Requirement Diagram
- Structural Diagram
- Block Definition Diagram
- Internal Block Diagram

- Package Diagram

### **Working with SysML**

- Structure: Definition and Use
- Behavior: Interaction, State Machine and activity/functions
- Requirements
- Parametrics
- SysML Diagram Frames
- Package Diagram
- Views
- Internal Block Diagram
- Allocations
- Basic Structural elements
- SysML Language Architecture
- Design Principles
- Architecture
- Extension Mechanisms
- SysML Diagrams

### **Structural Constructs**

- Model Elements
  - Overview
  - Diagram Elements
  - UML Extensions
  - Usage Examples
- Blocks
  - Overview
  - Diagram Elements
  - UML Extensions
  - Diagram Extensions
  - Stereotypes
  - Model Libraries
  - Usage Examples
- Ports and Flows
  - Standard Ports
  - Flow Ports

- Item Flows
  - Diagram Elements
    - UML Extensions
    - Diagram Extensions
    - FlowPort
    - FlowProperty
    - FlowSpecification
    - ItemFlow
    - StandardPort
    - Stereotypes
    - Block
    - FlowDirection
    - FlowPort
    - FlowProperty
    - FlowSpecification
    - ItemFlow
  - Constraint Blocks
    - Overview
    - Diagram Elements
    - UML Extensions
    - Diagram Extensions
    - Stereotypes
- Diagram Elements
  - Profiles & Model Libraries

### **Behavioral Constructs**

- Activities
- Diagram Elements
- UML Extensions
- Interactions
- Overview
- Diagram Elements
- State Machines
- Use Cases

### **Crosscutting Constructs**

- Allocations
- Requirements