



# SysML

## DISTILLED

A BRIEF GUIDE TO  
THE SYSTEMS  
MODELING LANGUAGE

LENNY DELLIGATTI

*Forewords by* RICK STEINER  
and RICHARD SOLEY

OMG SysML™

## Praise for *SysML Distilled*

“In keeping with the outstanding tradition of Addison-Wesley’s technical publications, Lenny Delligatti’s *SysML Distilled* does not disappoint. Lenny has done a masterful job of capturing the spirit of OMG SysML as a practical, standards-based modeling language to help systems engineers address growing system complexity. This book is loaded with matter-of-fact insights, starting with basic MBSE concepts to distinguishing the subtle differences between use cases and scenarios to illumination on namespaces and SysML packages, and even speaks to some of the more esoteric SysML semantics such as token flows.”

— Jeff Estefan, *Principal Engineer, NASA’s Jet Propulsion Laboratory*

“The power of a modeling language, such as SysML, is that it facilitates communication not only within systems engineering but across disciplines and across the development life cycle. Many languages have the potential to increase communication, but without an effective guide, they can fall short of that objective. In *SysML Distilled*, Lenny Delligatti combines just the right amount of technology with a common-sense approach to utilizing SysML toward achieving that communication. Having worked in systems and software engineering across many domains for the last 30 years, and having taught computer languages, UML, and SysML to many organizations and within the college setting, I find Lenny’s book an invaluable resource. He presents the concepts clearly and provides useful and pragmatic examples to get you off the ground quickly and enables you to be an effective modeler.”

— Thomas W. Fagnoli, *Lead Member of the Engineering Staff, Lockheed Martin*

“This book provides an excellent introduction to SysML. Lenny Delligatti’s explanations are concise and easy to understand; the examples well thought out and interesting.”

— Susanne Sherba, *Senior Lecturer, Department of Computer Science, University of Denver*

“Lenny hits the thin line between a reference book for SysML to look up elements and an entertaining book that could be read in its entirety to learn the language. A great book in the tradition of the famous *UML Distilled*.”

— Tim Weilkiens, *CEO, oose*

# SysML Distilled: A Brief Guide to the Systems Modeling Language

## Table of Contents

Cover

Half Title

Title Page

Copyright Page

Contents

Foreword

Foreword

Preface

Acknowledgments

About the Author

Chapter 1 Overview of Model-Based Systems  
Engineering

1.1 What Is MBSE?

1.2 The Three Pillars of MBSE

1.3 The Myth of MBSE

Chapter 2 Overview of the Systems Modeling  
Language

2.1 What SysML Is and Isn't

2.2 Yes, SysML Is Based on UML but You Can Start with  
SysML

2.3 SysML Diagram Overview

# **Table of Contents**

## 2.4 General Diagram Concepts

## **Chapter 3 Block Definition Diagrams**

### 3.1 Purpose

### 3.2 When Should You Create a BDD?

### 3.3 The BDD Frame

### 3.4 Blocks

### 3.5 Associations: Another Notation for a Property

### 3.6 Generalizations

### 3.7 Dependencies

### 3.8 Actors

### 3.9 Value Types

### 3.10 Constraint Blocks

### 3.11 Comments

## **Chapter 4 Internal Block Diagrams**

### 4.1 Purpose

### 4.2 When Should You Create an IBD?

### 4.3 Blocks, Revisited

### 4.4 The IBD Frame

### 4.5 BDDs and IBDs: Complementary Views of a Block

### 4.6 Part Properties

### 4.7 Reference Properties

### 4.8 Connectors

### 4.9 Item Flows

### 4.10 Nested Parts and References

## **Chapter 5 Use Case Diagrams**

### 5.1 Purpose

### 5.2 When Should You Create a Use Case Diagram?

# **Table of Contents**

5.3 Wait! Whats a Use Case?

5.4 The Use Case Diagram Frame

5.5 Use Cases

5.6 System Boundary

5.7 Actors

5.8 Associating Actors with Use Cases

5.9 Base Use Cases

5.10 Included Use Cases

5.11 Extending Use Cases

## **Chapter 6 Activity Diagrams**

6.1 Purpose

6.2 When Should You Create an Activity Diagram?

6.3 The Activity Diagram Frame

6.4 A Word about Token Flow

6.5 Actions: The Basics

6.6 Object Nodes

6.7 Edges

6.8 Actions, Revisited

6.9 Control Nodes

6.10 Activity Partitions: Allocating Behaviors to Structures

## **Chapter 7 Sequence Diagrams**

7.1 Purpose

7.2 When Should You Create a Sequence Diagram?

7.3 The Sequence Diagram Frame

7.4 Lifelines

7.5 Messages

7.6 Destruction Occurrences

# **Table of Contents**

7.7 Execution Specifications

7.8 Constraints

7.9 Combined Fragments

7.10 Interaction Uses

## **Chapter 8 State Machine Diagrams**

8.1 Purpose

8.2 When Should You Create a State Machine Diagram?

8.3 The State Machine Diagram Frame

8.4 States

8.5 Transitions

8.6 Pseudostates

8.7 Regions

## **Chapter 9 Parametric Diagrams**

9.1 Purpose

9.2 When Should You Create a Parametric Diagram?

9.3 Blocks, Revisited

9.4 The Parametric Diagram Frame

9.5 Constraint Properties

9.6 Constraint Parameters

9.7 Value Properties

9.8 Binding Connectors

## **Chapter 10 Package Diagrams**

10.1 Purpose

10.2 When Should You Create a Package Diagram?

10.3 The Package Diagram Frame

10.4 Notations for Namespace Containment

10.5 Dependencies between Packages

# **Table of Contents**

10.6 Importing Packages

10.7 Specialized Packages

10.8 Shades of Gray: Are You Looking at a Package  
Diagram or a Block Definition Diagram?

## **Chapter 11 Requirements Diagrams**

11.1 Purpose

11.2 When Should You Create a Requirements Diagram?

11.3 The Requirements Diagram Frame

11.4 Requirements

11.5 Requirements Relationships

11.6 Notations for Requirements Relationships

11.7 Rationale

## **Chapter 12 Allocations: Cross-Cutting Relationships**

12.1 Purpose

12.2 There's No Such Thing as an Allocation Diagram

12.3 Uses for Allocation Relationships

12.4 Notations for Allocation Relationships

12.5 Rationale

**Appendix A: SysML Notation Desk Reference**

**Appendix B: Changes between SysML Versions**

**Bibliography**

**Index**

**Blank Page**