

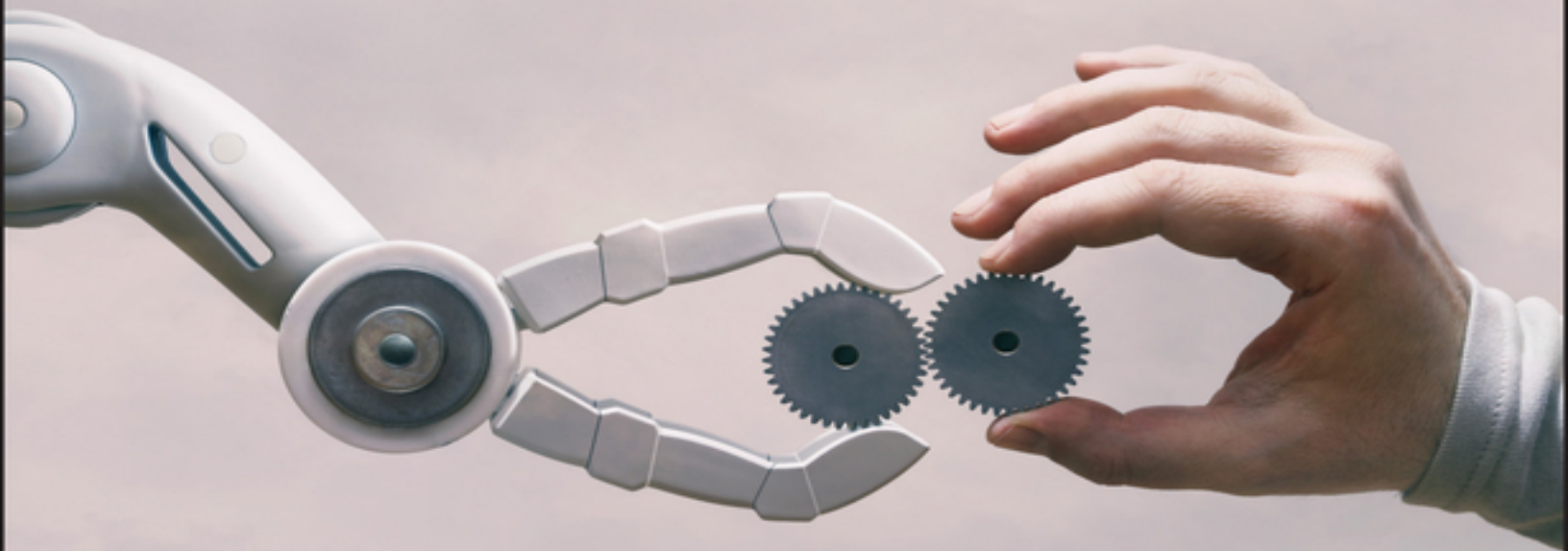
Wiley Series in Systems Engineering and Management

William Rouse, Series Editor

SECOND EDITION

# MODEL-BASED SYSTEM ARCHITECTURE

TIM WEILKIENS • JESKO G. LAMM  
STEPHAN ROTH • MARKUS WALKER



WILEY

# Table of Contents

[Cover](#)

[Title Page](#)

[Copyright](#)

[Foreword](#)

[Preface](#)

[About the Companion Website](#)

[1 Introduction](#)

[2 An Example: The Scalable Observation and Rescue System](#)

[3 Better Products - The Value of Systems Architecting](#)

[3.1 The Share of Systems Architecting in Making Better Products](#)

[3.2 Benefits that can be Achieved](#)

[3.3 Benefits that can be Communicated Inside the Organization](#)

[3.4 Beneficial Elements of Systems Architecting](#)

[3.5 Benefits of Model-Based Systems Architecting](#)

[4 Systems, Systems of Systems, and Cyber-Physical Systems\\*](#)

[4.1 Definition of "System"](#)

[4.2 Definition of "System of Systems"](#)

[4.3 Definition of "Cyber-Physical System"](#)

[4.4 Composition of a "Cyber-Physical System of Systems"](#)

[Note](#)

[5 Definition of System Architecture](#)

[5.1 What Is Architecture? – Discussion of Some Existing Definitions](#)

[5.2 Relations Between Concepts of “System,” “Architecture,” and “Architecture Description”](#)

[5.3 Definition of “Architecture”](#)

[5.4 Functional and Physical Architecture](#)

[5.5 Taxonomy of Physical Architectures](#)

[5.6 Architecture Landscape for Systems](#)

[6 Model-Based Systems Architecting](#)

[Notes](#)

[7 Model Governance](#)

[7.1 Overview](#)

[7.2 Model Governance in Practice](#)

[Note](#)

[8 Architecture Description](#)

[8.1 Architecture Descriptions for Stakeholders](#)

[8.2 Definition of “Architecture Description”](#)

[8.3 How to Get Architecture Descriptions?](#)

[Notes](#)

[9 Architecture Patterns and Principles](#)

[9.1 The SYSMOD Zigzag Pattern](#)

[9.2 The Base Architecture](#)

[9.3 Cohesion and Coupling](#)

[9.4 Separation of Definition, Usage, and Run-Time](#)

[9.5 Separate Stable from Unstable Parts](#)

[9.6 The Ideal System](#)

[9.7 View and Model](#)

[9.8 Diagram Layout](#)

[9.9 System Model Structure](#)

[9.10 System Architecture Principles](#)

[9.11 Heuristics](#)

[Note](#)

[10 Model-Based Requirements Engineering and Use Case Analysis](#)

[10.1 Requirement and Use Case Definitions](#)

[10.2 Model-Based Requirements and Use Case Analysis from the MBSA Viewpoint](#)

[10.3 The SAMS Method](#)

[10.4 Use Cases 2.0](#)

[11 Perspectives, Viewpoints and Views in System Architecture](#)

[11.1 Introduction](#)

[11.2 The Functional Perspective](#)

[11.3 The Physical Perspective](#)

[11.4 The Behavioral Perspective](#)

[11.5 The Layered Perspective](#)

[11.6 System Deployment Perspective](#)

[11.7 Other Perspectives](#)

[11.8 Relation to the System Context](#)

[11.9 Mapping Different System Elements Across Different Levels](#)

[11.10 Traceability](#)

[11.11 Perspectives and Architecture Views in Model-based Systems Architecting](#)

[Notes](#)

[12 Typical Architecture Stakeholders](#)

[12.1 Overview](#)

[12.2 Requirements Engineering](#)

[12.3 Verification](#)

[12.4 Configuration Management](#)

[12.5 Engineering and Information Technology Disciplines](#)

[12.6 Project and Product Management](#)

[12.7 Risk Managers](#)

[12.8 Development Roadmap Planners](#)

[12.9 Production and Distribution](#)

[12.10 Suppliers](#)

[12.11 Marketing and Brand Management](#)

[12.12 Management](#)

[Notes](#)

[13 Roles](#)

[13.1 Roles](#)

[13.2 The System Architect Role](#)

[13.3 System Architecture Teams](#)

[13.4 System Architecture Stakeholders](#)

[13.5 Recruiting System Architecture People](#)

[13.6 Talent Development for System Architects](#)

[14 Processes](#)

[14.1 Systems Architecting Processes](#)

[14.2 Design Definition Process](#)

[14.3 Change and Configuration Management Processes](#)

[14.4 Other Processes Involving the System Architect](#)

[Notes](#)

[15 Tools for the Architect](#)

[16 Agile Approaches](#)

[16.1 The History of Iterative-Incremental Approaches](#)

[16.2 The Manifesto for Agile Software Development \(2001\)](#)

[16.3 Agile Principles in Systems Engineering](#)

[16.4 Scaling Agile](#)

[16.5 System Architects in an Agile Environment](#)

[17 The FAS Method](#)

[17.1 Motivation](#)

[17.2 Functional Architectures for Systems](#)

[17.3 How the FAS Method Works](#)

[17.4 FAS Heuristics](#)

[17.5 FAS with SysML](#)

[17.6 SysML Modeling Tool Support](#)

[17.7 Mapping of a Functional Architecture to a Physical Architecture](#)

[17.8 Experiences with the FAS Method](#)

[17.9 FAS Workshops](#)

[17.10 Quality Requirements and the Functional Architecture](#)

[17.11 Functional Architectures and the Zigzag Pattern](#)

[17.12 CPS-FAS for Cyber-physical Systems](#)

[Notes](#)

[18 Product Lines and Variants](#)

[18.1 Definitions Variant Modeling](#)

[18.2 Variant Modeling with SysML](#)

[18.3 Other Variant Modeling Techniques](#)

[19 Architecture Frameworks](#)

[19.1 Enterprise Architectures](#)

[19.2 Characteristics of System of Systems \(SoS\)](#)

[19.3 An Overview of Architecture Frameworks](#)

[19.4 System Architecture Framework \(SAF\)](#)

[Together with Michael Leute](#)

[19.5 What to Do When We Come in Touch With Architecture Frameworks](#)

[Notes](#)

[20 Cross-cutting Concerns](#)

[20.1 The Game-Winning Nonfunctional Aspects](#)

[20.2 Human System Interaction and Human Factors Engineering](#)

[20.3 Risk Management](#)

[20.4 Trade Studies](#)

[20.5 Budgets](#)

[21 Architecture Assessment](#)

[Note](#)

[22 Making It Work in the Organization](#)

[22.1 Overview](#)

[22.2 Organizational Structure for Systems Architecting](#)

[22.3 Recipes from the Authors' Experience](#)

[23 Soft Skills](#)

[23.1 It's All About Communication](#)

[23.2 Personality Types](#)

[23.3 Team Dynamics](#)

[23.4 Diversity and Psychological Safety](#)

[23.5 Intercultural Collaboration Skills](#)

[Notes](#)

[24 Outlook: The World After Artificial Intelligence](#)

[Appendix A OMG Systems Modeling Language](#)

[A.1 Architecture of the Language](#)

[A.2 Diagram and Model](#)

[A.3 Structure Diagrams](#)

[A.4 Behavior Diagrams](#)

[A.5 Requirements Diagram](#)

[A.6 Extension of SysML with Profiles](#)

[A.7 Next-Generation Modeling Language SysML v2](#)

[Appendix B The V-Model](#)

[B.1 A Brief History of the V-Model or the Systems Engineering Vee](#)

[B.2 A Handy Illustration but No Comprehensive Process Description](#)

[B.3 Critical Considerations](#)

[B.4 Reading Instruction for a Modern Systems Engineering Vee](#)

[Appendix C Glossary](#)

[C.1 Heritage of the Term “Glossary”](#)

[C.2 Terms with Specific Meaning](#)

[References](#)

[Index](#)

[Wiley End User License Agreement](#)

## **List of Tables**

Chapter 8

[Table 8.1 Explanation of content of architecture description template for m...](#)



## Chapter 9

[Table 9.1 Brief description of base architecture elements of the Virtual To...](#)

## Chapter 11

[Table 11.1 Modular vs. integral Architectures according to Ulrich \[253\]](#)

[Table 11.2 A functional-to-physical allocation matrix between the functiona...](#)

## Chapter 12

[Table 12.1 Close collaboration between system architects and requirements p...](#)

[Table 12.2 Close collaboration between system architects and verification p...](#)

[Table 12.3 Close collaboration between system architects and configuration ...](#)

[Table 12.4 Close collaboration between system architects and subject matter...](#)

[Table 12.5 Close collaboration between system architects and project leader...](#)

[Table 12.6 Close collaboration between system architects and roadmap planne...](#)

[Table 12.7 Close collaboration between system architects and production and...](#)

[Table 12.8 Close collaboration between system architects and the supplier's...](#)

[Table 12.9 Close collaboration between system architects and marketing or b...](#)

[Table 12.10 A good relationship between system architects and management as...](#)

## Chapter 14

[Table 14.1 Examples of inputs, outputs, and contributors for the systems ar...](#)

## Chapter 17

[Table 17.1 Mapping of FAS concepts to SysML](#)

## Chapter 23

[Table 23.1 The four sides of the message in the communication depicted in Fi...](#)

[Table 23.2 The eight psychological types by C. G. Jung.](#)

# List of Illustrations

## Chapter 2

[Figure 2.1 The museum tour robot.](#)

[Figure 2.2 Storyboard explaining the Virtual Museum Tour system.](#)

[Figure 2.3 The Scalable Observation and Rescue System supporting a rescue mi...](#)

## Chapter 3

[Figure 3.1 Systems architecting plays a major role in breaking down requirem...](#)

## Chapter 4

[Figure 4.1 Definition of "System."](#)

[Figure 4.2 Entities playing the role of system elements are regarded either ...](#)

[Figure 4.3 System characteristics are the generalization of properties, beha...](#)

[Figure 4.4 Sierpinski triangle representing self-similarity in mathematics....](#)

[Figure 4.5 An SoS representation of the rescue system described in Chapter 2...](#)

[Figure 4.6 Cyber-physical Systems of Systems are special types of Systems of...](#)

## Chapter 5

[Figure 5.1 Relations between concepts of system, architecture, architecture ...](#)

[Figure 5.2 Definition of “Architecture.”](#)

[Figure 5.3 Architecture evolves through allocation of functional elements to ...](#)

[Figure 5.4 Specialization of physical architecture into logical architecture...](#)

[Figure 5.5 Architecture landscape for systems.](#)

[Figure 5.6 Definition of system architecture.](#)

## Chapter 6

[Figure 6.1 MBSA ontology.](#)

[Figure 6.2 System model.](#)

[Figure 6.3 MDA levels.](#)

[Figure 6.4 SYSMOD intensity model.](#)

## Chapter 8

[Figure 8.1 Impacts of any kind on processes participated by stakeholders cau...](#)

[Figure 8.2 The purpose is a kind of concern.](#)

[Figure 8.3 Definition of “Architecture Description.”](#)

[Figure 8.4 Definition of “Architecture Viewpoint.”](#)

[Figure 8.5 Definition of “Architecture View.”](#)

[Figure 8.6 Definition of “Architecture Decision.”](#)

[Figure 8.7 Definition of “Architecture Rationale.”](#)

[Figure 8.8 Headline structure of the Architecture Template 1.0 by the workin...](#)

## Chapter 9

[Figure 9.1 The SYSMOD zigzag\\_pattern.](#)

[Figure 9.2 Relationship between requirements and architecture kinds.](#)

[Figure 9.3 Zigzag\\_pattern in the V-Model.](#)

[Figure 9.4 One zigzag\\_pattern level in the model.](#)

[Figure 9.5 SysML rationale element to document derive relationship.](#)

[Figure 9.6 Definition of extended derive relationship.](#)

[Figure 9.7 Example of the extended derive relationship.](#)

[Figure 9.8 Model structure and relationships of the zigzag\\_pattern.](#)

[Figure 9.9 Strictly separated zigzag levels.](#)

[Figure 9.10 Base architecture of the Virtual Tour system.](#)

[Figure 9.11 Relationship of top level requirements to base architecture.](#)

[Figure 9.12 Product tree of the Virtual Tour base architecture.](#)

[Figure 9.13 Virtual Tour system constraint requirement for base architecture...](#)

[Figure 9.14 Cohesion and coupling.](#)

[Figure 9.15 Definition of a screw with SysML.](#)

[Figure 9.16 Usage of a screw with SysML.](#)

[Figure 9.17 Run-time setting of the screw with SysML.](#)

[Figure 9.18 Internal structure of the screw.](#)

[Figure 9.19 Abstract syntax of a use case.](#)

[Figure 9.20 Different layout directions.](#)

[Figure 9.21 Top level system model structure.](#)

[Figure 9.22 Example Principle 2 Down, 1 Up.](#)

## Chapter 10

[Figure 10.1 Requirement ontology.](#)

[Figure 10.2 Use case ontology.](#)

[Figure 10.3 Requirements in a SysML model.](#)

[Figure 10.4 Virtual tour system context.](#)

[Figure 10.5 Extended system context.](#)

[Figure 10.6 System use cases.](#)

[Figure 10.7 Continuous use case “Charge robot battery.”](#)

[Figure 10.8 Table view for use cases.](#)

[Figure 10.9 Use case activity “Book a tour.”](#)

[Figure 10.10 Example activity diagram \*Monitor and charge robot battery.\*](#)

[Figure 10.11 Extract of the Virtual Tour domain knowledge model.](#)

[Figure 10.12 SAMS method ontology.](#)

[Figure 10.13 Virtual tour personas.](#)

[Figure 10.14 Storyboard image “Select tour”.](#)

[Figure 10.15 Use Case 2.0 Ontology.](#)

[Figure 10.16 Use case activity “Steer the robot - human.”](#)

## Chapter 11

[Figure 11.1 Excerpt from the functional decomposition of the Virtual Tour sy...](#)

[Figure 11.2 Excerpt from the functional decomposition of the Virtual Tour sy...](#)

[Figure 11.3 An architecture view from the functional perspective for a funct...](#)

[Figure 11.4 Example architecture view on the Virtual Tour system in the logi...](#)

[Figure 11.5 A data view related to the logical architecture as the basis of ...](#)

[Figure 11.6 Example architecture view on the Virtual Tour system from the ph...](#)

[Figure 11.7 Example architecture view on the Virtual Tour system from the ph...](#)

[Figure 11.8 An informal architecture view from the layered perspective.](#)

[Figure 11.9 Layer-independent domain block “Position” and the derived layer-...](#)

[Figure 11.10 Example architecture view on the Virtual Tour system from the l...](#)

[Figure 11.11 An interface view from the layered perspective: details on inte...](#)

[Figure 11.12 SysML representation of the layer stacks on different devices, ...](#)

[Figure 11.13 SysML representation of the layer stacks on different devices, ...](#)

[Figure 11.14 An architecture view from the system deployment perspective in ...](#)

[Figure 11.15 A constraint in the system deployment perspective in informal r...](#)

[Figure 11.16 The constraint from Figure 11.15 in the system deployment persp...](#)

[Figure 11.17 Example view on the Virtual Tour system, taking parts of the sy...](#)

[Figure 11.18 An informal representation of the verification context.](#)

[Figure 11.19 The verification context specification via a system context dia...](#)

[Figure 11.20 System design and its interconnection and overlap with system e...](#)

[Figure 11.21 Logical-to-product mapping in a communication-focused view.](#)

[Figure 11.22 The partially hand-made version of Figure 11.5, as it could hav...](#)

## Chapter 12

[Figure 12.1 Collaboration between requirements engineering \(RE\) and systems ...](#)

[Figure 12.2 © 2014 Jakob K., reproduced with permission.](#)

[Figure 12.3 A fictitious example of predicting architecture workload over ti...](#)

[Figure 12.4 Split of concerns between system architect, project manager, and...](#)

[Figure 12.5 Using workload predictions from Figure 12.3 to support planning ...](#)

## Chapter 13

[Figure 13.1 Abstraction skills.](#)

[Figure 13.2 Changing profile of a system architect with growing experience,...](#)

## Chapter 16

[Figure 16.1 Mercury's Friendship 7 capsule at the National Air and Space Mus...](#)

[Figure 16.2 Adaptation and simplified representation of the Spiral Model acc...](#)

[Figure 16.3 The development process according to the Scrum framework.](#)

## Chapter 17

[Figure 17.1 Functional-to-physical mapping.](#)

[Figure 17.2 Domain model for functional architectures.](#)

[Figure 17.3 Matrix functional grouping.](#)

[Figure 17.4 Relationship between functional groups and functional elements....](#)

[Figure 17.5 Part of the functional architecture of the TourBot.](#)

[Figure 17.6 Definitions of some interface blocks.](#)

[Figure 17.7 Example: Parts of the Activity Tree of the use case "Steer the r...](#)



[Figure 17.8 Traceability between functional blocks and functional groups.](#)

[Figure 17.9 Traceability from functional blocks to use case activities.](#)

[Figure 17.10 Function structure of the virtual tour system.](#)

[Figure 17.11 Example functional architecture of the virtual tour system.](#)

[Figure 17.12 Types of the functional block ports.](#)

[Figure 17.13 FAS process.](#)

[Figure 17.14 Manual and automated steps of the FAS task “Identify Functional...”](#)

[Figure 17.15 Manual and automated steps of the FAS task “Model Functional Ar...”](#)

[Figure 17.16 Example use case activity “Book a Tour.”](#)

[Figure 17.17 Example use case activity “Identify Customer.”](#)

[Figure 17.18 Functional to physical mapping matrix.](#)

[Figure 17.19 Example of use case cards.](#)

[Figure 17.20 Example of a activity card.](#)

[Figure 17.21 Relationship of functional and quality requirements.](#)

[Figure 17.22 Traceability from functional architecture to nonfunctional requ...](#)

[Figure 17.23 Nonfunctional requirements in the functional architecture.](#)

[Figure 17.24 Example FAS in the zigzag\\_pattern - level <n>.](#)

[Figure 17.25 Example FAS in the zigzag\\_pattern - level <n+1>.](#)

[Figure 17.26 CPS use cases for human and machine.](#)

[Figure 17.27 Aggregated system.](#)

[Figure 17.28 Aggregated system use case.](#)

[Figure 17.29 Aggregated system use case activity.](#)

[Figure 17.30 Aggregated system functional architecture.](#)

## Chapter 18

[Figure 18.1 Domain model for variant modeling.](#)

[Figure 18.2 Core and variants.](#)

[Figure 18.3 Top level package structure.](#)

[Figure 18.4 SysML feature tree with variants.](#)

[Figure 18.5 Relationship between variant and core elements.](#)

[Figure 18.6 Variant configuration.](#)

[Figure 18.7 Example FODA Tree.](#)

[Figure 18.8 Example OVM.](#)

## Chapter 19

[Figure 19.1 The interrelationship between business goals, information archit...](#)

[Figure 19.2 The TOGAF® architecture development method \(ADM\).](#)

[Figure 19.3 Enterprise architecture frameworks and system architecture frame...](#)

## Chapter 21

[Figure 21.1 Process steps of ATAM.](#)

[Figure 21.2 Virtual Tour system objectives.](#)

[Figure 21.3 Excerpt from the quality attributes utility tree for the irtual ...](#)

## Chapter 22

[Figure 22.1 Examples showing different alternatives for anchoring systems ar...](#)

[Figure 22.2 Example showing the rating of how well organizational interfaces...](#)

## Chapter 23

[Figure 23.1 A visualization of the four-sides model.](#)

[Figure 23.2 What are the four sides of the message in this communication?...](#)

[Figure 23.3 The Allen Curve depicts the negative correlation between frequen...](#)

## Appendix A

[Figure A.1 SysML diagram types.](#)

[Figure A.2 UML for SysML.](#)

[Figure A.3 SYSMOD, SysML, and UML.](#)

[Figure A.4 Diagram interchange.](#)

[Figure A.5 SysML structure diagram types.](#)

[Figure A.6 SysML block.](#)

[Figure A.7 Example Product Tree of the Virtual Tour system.](#)

[Figure A.8 Example domain model of the virtual tour system.](#)

[Figure A.9 Activity tree.](#)

[Figure A.10 Activity tree with associated blocks.](#)

[Figure A.11 Example internal block diagram of the virtual tour system.](#)

[Figure A.12 Flow properties and provided/requested features.](#)

[Figure A.13 Full and proxy port.](#)

[Figure A.14 Behavioral proxy ports in the functional architecture.](#)

[Figure A.15 Interface block specialization.](#)

[Figure A.16 Definition of a constraint property.](#)

[Figure A.17 Example of a parametric diagram.](#)

[Figure A.18 Model browser of a modeling tool.](#)

[Figure A.19 Example package diagram of the VMT.](#)

[Figure A.20 Tree-like notation of nested packages.](#)

[Figure A.21 SysML behavior diagram types.](#)

[Figure A.22 Example use case diagram.](#)

[Figure A.23 Example use case subject and include relationship.](#)

[Figure A.24 Example activity diagram.](#)

[Figure A.25 Opaque actions.](#)

[Figure A.26 Accept event action and send signal action.](#)

[Figure A.27 Example state machine diagram.](#)

[Figure A.28 Example sequence diagram.](#)

[Figure A.29 Example requirements diagram.](#)

[Figure A.30 Example requirements table.](#)

[Figure A.31 Example requirements matrix.](#)

[Figure A.32 Stereotype WeightedSatisfy.](#)

[Figure A.33 Graphical notation of the SysML v2 sample model.](#)

[Figure A.34 SysML v2 API and Services concept.](#)

## Appendix B

[Figure B.1 A basic V-Model.](#)

[Figure B.2 V-Model with exemplary named basic development processes.](#)

[Figure B.3 V-Model considering discrete levels and numbers of elements.](#)

## Appendix C

[Figure C.1 A page of an eleventh century manuscript with glosses.](#)

**WILEY SERIES IN SYSTEMS ENGINEERING AND  
MANAGEMENT**

**William Rouse, Series Editor**  
**Andrew P. Sage, Founding Editor**

A complete list of the titles in this series appears at the end of this volume.

# **Model-Based System Architecture**

Second Edition

*Tim Weilkiens*

Hamburg, Germany

*Jesko G. Lamm*

Bern, Switzerland

*Stephan Roth*

Hamburg, Germany

*Markus Walker*

Ziefen, Switzerland

# **WILEY**

This edition first published 2022  
© 2022 John Wiley & Sons, Inc.

*Edition History*

John Wiley & Sons (1e, 2015)

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by law. Advice on how to obtain permission to reuse material from this title is available at <http://www.wiley.com/go/permissions>.

The right of Tim Weilkiens, Jesko G. Lamm, Stephan Roth, and Markus Walker to be identified as the authors of this work has been asserted in accordance with law.

*Registered Office*

John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

*Editorial Office*

111 River Street, Hoboken, NJ 07030, USA

For details of our global editorial offices, customer services, and more information about Wiley products visit us at [www.wiley.com](http://www.wiley.com).

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

*Limit of Liability/Disclaimer of Warranty*

In view of ongoing research, equipment modifications, changes in governmental regulations, and the constant flow of information relating to the use of medicines, equipment, and devices, the reader is urged to review and evaluate the information provided in the package insert or instructions for each medicine, equipment, or device for, among other things, any changes in the instructions or indication of usage and for added warnings and precautions. While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor



authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

*Library of Congress Cataloging-in-Publication Data applied for:*  
ISBN: 9781119746652

Cover Design: Wiley

Cover Image: © Ivan Bajic/Getty Images

# Foreword

Contrary to popular myth, models are not new to systems engineering. Models are the way engineers analyze both problems and solutions, so systems models are as old as systems engineering itself. With the traditional focus on written specifications as the “source of truth,” models were secondary and descriptive – sometimes reflected as simple sketches, sometimes shown in formal diagrams, partially captured in analysis packages, and often trapped in the mind of the chief engineer. The transformation of systems engineering from document-centric to model-centric practices is not about the introduction of models. It is about making models explicit and moving them to the foreground where they serve as the authoritative tool for design, analysis, communication, and system specification.

Organizations today are investing heavily in representations, standards, methodologies, and technologies to transform the practice of systems engineering through model-driven paradigms. To manage the complexity of today's problems; to keep pace with today's rapidly evolving technologies; to capture the required knowledge regarding the problem, solution, and rationale; to respond effectively to change – all require that systems engineering join the other engineering disciplines in moving beyond document-centric techniques and embracing the power of a model-based foundation. With energy and focus over the last 10 years has come notable progress. The industry has advanced in the area of representations with the development of SysML as a standardized set of diagrams to complement traditional systems representations. Numerous books – including a frequently-cited guide by Tim Weilkiens – explain the details

of using this notation to capture and communicate system designs to improve explicitness and alignment within the systems team. Alongside these representations have emerged countless standards and frameworks to help engineering teams develop high fidelity models reflecting key systems dimensions.

However, for all the industry discussion regarding SysML, representations, standards, and tools, there remains a great deal of confusion. Understanding SysML notation and drawing SysML diagrams do not equate to doing model-based systems engineering. Nor is the use of disjoint models and simulation in systems engineering equivalent to integrated model-based systems engineering.

Effectively moving forward with the transition to model-centric techniques requires that we step back to understand the bigger picture. Diagrams and other representations do not live in isolation but are interrelated and overlapping, communicating key aspects of the system model from specific viewpoints. System architecture and detailed analytical models are not disjoint, nor is there a single grand unified model to capture all dimensions of interest for all systems problems. To move forward, we must embrace the holistic systems perspective and apply it to model-based systems engineering, seeking out the interrelationships and developing a robust toolbox of supporting practices.

In this book, Tim Weilkiens, Jesko Lamm, Stephan Roth, and Markus Walker broaden our vision and expose us to a rich set of perspectives, processes, and methods so that we can develop an effective unified framework for model-based systems architecture. Building upon the existing industry library of textbooks on SysML, this book looks beyond representation to address models, viewpoints, and views as part of a modern approach addressing requirements,

behavior, architecture, and more. It connects to a larger framework of processes, methods, and tools key to enabling model-centric practices. And it looks beyond the technical space to the critical cultural dimensions, because the transformation to model-centric techniques is far less a technical challenge than one of organizational change. Addressing the broader framework, Tim, Jesko, Stephan, and Markus bring model-centric practices together to help practitioners develop cohesive system architectures - our one chance in the life of a program to manage complexity, develop resilience, and design in critical concerns such as system security.

There is no doubt that the future of systems engineering is model-based. Document-centric techniques simply are not enough as we grapple with the challenges of today and tomorrow. Those practitioners and organizations who are early adopters in developing a cohesive model-centric framework of processes, methods, and tools will certainly be at a competitive advantage - whether producing products themselves or delivering systems services for others. If, as a profession, we can transform from document-centric to model-based systems engineering and do so with the vision of enabling model-based engineering, we can help transform the larger product lifecycle delivering radical improvements in quality, cost, and time-to-market for the benefit of all.

<i>June 2015</i>	<i>David Long President, Vitech Corporation INCOSE President (2014 and 2015)</i>
------------------	--

# Preface

Reacting to market needs on time with systems of high quality and marketable costs is a strong competitive advantage. Once a market need has been identified, multiple disciplines are involved in developing a system toward it. They need to collaborate closely and each according to a precise understanding of the own contribution to the system development. Effective communication and the creation of understanding for the whole system-of-interest are keys for the success. Organizations are facing a more and more dynamic environment and, at the same time, an increasing organizational complexity of distributed teams and stakeholders and an increasing technical complexity of more heterogeneous relationships between system components and their environment. This context requests an explicit and sustainable system architecture.

Each of the engineering disciplines contributing to system development needs specific views for obtaining the needed insight. System models enable the creation of consistent sets of stakeholder-specific views. People using them gain a fast and comprehensible understanding of the system they are developing, which can help them choose appropriate solutions for fulfilling the market needs. All the views look at the same data baseline. There is no effort to consolidate redundant data or to clarify misunderstandings of inconsistent information and the costs of resulted errors.

A system architect needs to shape the system architecture well for realizing a successful system. Multiple tasks have to be carried out, each using an effective approach. This book provides a toolbox for the architects for their daily challenges. The scope of the book is a model-based

environment, that is either already established and running or planned. The book explains how to use the SysML modeling language in obtaining model-based architecture descriptions. Nevertheless, the concepts are independent of SysML and could also be performed with other modeling languages.

This book is about people, models, and better products, based on our belief that model-based systems architecting produces better products by creating communication and insight for people involved in system development. The book presents a collection of methods and approaches, which we see as ingredients for getting the system architecture work done successfully. We present model-based systems architecting, which we see as a required backbone for excellent system architecture work together with the stakeholders. We will show that involving the stakeholders means much more than running through a formalized review process.

A fundamental principle in system architecture is simplification. Without simple concepts to be communicated to the stakeholders, the system architect will not be understood and thus will fail. We advise you, dear reader, to adopt the principle of simplification and apply it to the multitude of approaches presented in the book. Feel free only to choose the most suitable approaches for your daily work and disregard the others until you are in a situation where they turn out to be the useful ones. The book is a well-stocked toolbox and not a rigid all-or-nothing process for system architects.

Our experience tells us that each organization will have a different focus area and will need different approaches. This is why we have bundled a variety of approaches we have observed being applied successfully in the industry, in the hope that you will find some pieces of information that