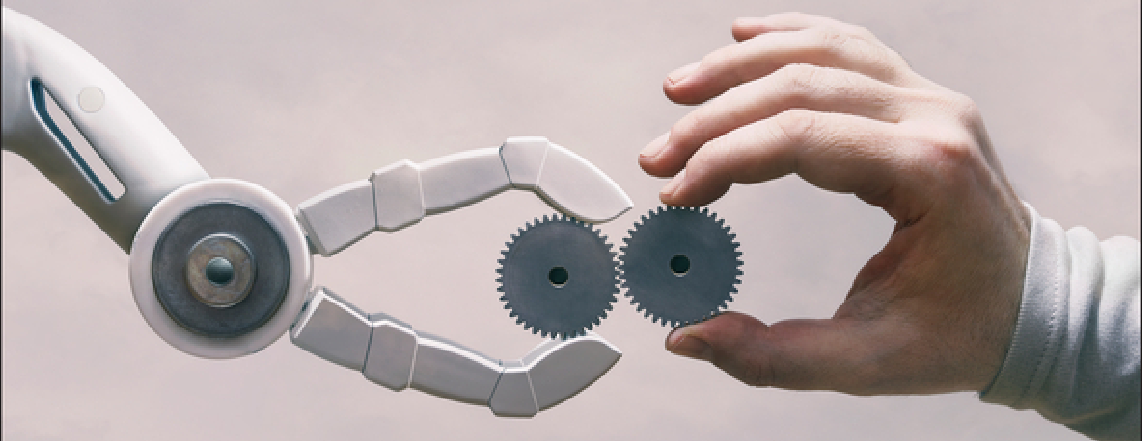
Wiley Series in Systems Engineering and Management
William Rouse, Series Editor

SECOND EDITION

MODEL-BASED SYSTEM ARCHITECTURE

TIM WEILKIENS • JESKO LAMM
STEPHAN ROTH • MARKUS WALKER



WILEY

Model-Based System Architecture

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

Edition History
John Wiley & Sons (1e, 2015)

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.

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.

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

Cover Design: Wiley
Cover Image: © Ivan Bajic/Getty Images

Set in 9.5/12.5pt STIXTwoText by Straive, Chennai, India

10 9 8 7 6 5 4 3 2 1

Contents

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.

June 2015

David Long
President, Vitech Corporation
INCOSE President (2014 and 2015)

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 are suitable exactly to your current activities. We have selected those approaches, which we find easy to apply in daily work and which are important for common model-based system architectures. We do not claim to provide a complete set. Every system architect loves to go to a hardware store to extend her toolbox. And from time to time she has to discard one of her tools when it is no longer appropriate.

The book addresses system architects and their managers as well as engineers who are involved or interested in systems architecting. It is the first comprehensive book that combines the emergent discipline systems architecting with model-based approaches with SysML and puts together puzzle pieces to a complete picture. Highlighted puzzle pieces are:

- functional architectures and the Functional Architecture for Systems (FAS) method by Lamm and Weilkiens to derive the architecture from common-use case analysis
- the integration of the concept of layered architectures from the software discipline in the context of system architectures
- the modeling of system variants
- the whole picture of different architecture kinds like functional, logical, and product architectures and their relationships
- a brief description of SysML and
- a summary of the history of the V-model and recent thinking about it in the appendix

As a typical reader of this book, you may have no time to read all chapters in sequential order. Therefore, we have made the chapters as independent from each other as we could, in order to enable you to read them individually or out of a dedicated sequence when you like inspiration about a certain topic. You can find an on-demand reference about particular topics and get inspiration for directly using the presented approaches in your daily business. The topics are demonstrated using a fictitious robot-based solution for virtual exhibition or other robot-based telepresence tours as an example system.

We like to write texts using gender-fair language. On the other hand, we avoid cluttering the flow of reading by using always both genders in the same sentence. Therefore, we have only used one gender where it was not appropriate to use gender-neutral language. Feel free to replace "he" by "she" and "she" by "he" or whatever is appropriate.

We like to thank the "FAS" and "MkS" working groups of GfSE, German chapter of INCOSE, as well as the "Viewpoints" working group of the same chapter in collaboration with Swiss Society for Systems Engineering (SSSE), Swiss chapter of INCOSE. The work in these groups has provided us with new ideas that can now be found in this book. We thank NoMagic for their support in working with the Cameo tool family that we used to create the SysML models and diagrams we used in multiple chapters of this book. We also thank Erik Solda for allowing us to use the robot example, Martin Ruch for contributing ideas about the assessment of organizational interfaces, and all the colleagues at work who have influenced our way of thinking, helped us with foreign languages in both reading and writing or recommended literature and web links that are today part of the foundations of this book. We furthermore thank numerous people who provided us with advice after we had shown or explained them little fragments of this book to listen to a second opinion.

We like to thank all the supporters of MBSE who believe that MBSE enables the successful development of complex systems – in particular, David Long, who is a great expert of MBSE from the very beginning and has written the foreword.

Finally, we like to thank Brett Kurzman, editor at Wiley, Alex Castro, Kathleen Pagliaro, Bhargavi Natarajan, Sarah Lemore, and Viniprammia Premkumar for their support with the first and second editions of this book.

September 2021

Tim Weilkiens, Jesko G. Lamm, Stephan Roth, Markus Walker
Contributors: Matthias Dänzer, Oliver C. Eichmann,
Ralf God, Michael Leute, Sylvia Melzer

About the Companion Website

This book is accompanied by a companion website:

www.mbse-architecture.com

The website includes:

- High resolution version of all the figures in the book.

1

Introduction

Model-based system architecture (MBSA) combines the two key technologies model-based and systems architecting. Both are major parts of the future state of systems engineering [123].

Many systems result from a evolutionary development. They are driven by their parts and do not emerge from the architecture. The parts could be anything that, in combination, is assembled to a human-made purposeful system. System architecture is followed by a complete system. Often system architecture is referred to the architecture from the perspective of a software architecture in combination with the hardware or the architecture of software-intensive systems [43]. We understand system architecture more holistic and also consider systems without any software, even though systems without any software, are rarely handled with systems engineering processes and MBSA concepts like described in this book. A system architecture is always present. In today and future systems engineering, it is crucial to apply explicit systems architecting for the success of the system project [123]. Chapter 5 defines the term "system architecture" within its context.

Studies clearly show that systems architecting is critical for the performance and success of the system [68]. This is particularly evident for projects that require significant architectural work or rework. Due to more and more dynamic and complex markets and environments, the system architecture must more and more support the changing requirements and requests for radical changes. Chapter 3 lists the benefits of systems architecting.

A system architecture is about establishing solutions that are in line with the directions that guide the organization and checked for feasibility by the corresponding experts, about designing interfaces that are agreed from both sides, and about ensuring that the people who should know the architecture of a system have a common understanding of it. MBSA uses models for enabling the creation of healthy communication around the architecture of the system and for ensuring that the architecture is validated from different points of view. Models are a key

Model-Based System Architecture, Second Edition. Tim Weilkiens, Jesko G. Lamm, Stephan Roth, and Markus Walker.

tool to be capable of developing complex systems on time and in a feasible quality. Chapter 6 defines the term "model" and MBSA and discusses related terms.

Models are more than graphics. There are even models without any graphical representations. Just the graphics is not modeling but drawing. To create a model, you need the concrete syntax, the abstract syntax, and the semantics, which you find in a modeling language. We use the international standard Systems Modeling Language (SysML) as a language for the system requirements and architecture models. Appendix A gives an overview about SysML, including an outlook on the next-generation modeling language SysML v2. Although we extensively use SysML in this book, our methods and concepts are independent of SysML and could also be implemented by any other modeling language.

The system architect is the one in charge of shaping the system architecture. This is a big responsibility and a big challenge. Organizations developing systems should carefully select people who are allowed to architect the system – and these people's work results will be tightly monitored by stakeholders everywhere in the organization. Chapter 22 describes how systems architecting could be embedded in an organization, and Chapter 12 discusses the interfaces to the stakeholders of systems architecting. In particular, Chapter 10 introduces the adjacent discipline requirements engineering that closely collaborates with the systems architecting. The SYSMOD zigzag pattern presented in Chapter 9 shows the relationship between requirements and architecture and clearly demonstrates the need for a close collaboration. Artifacts of the model-based requirements and use case analysis are important inputs for the system architects, especially to elaborate a functional architecture using the so-called Functional Architectures for Systems (FAS) method.

Chapter 17 is a comprehensive presentation of the FAS method. Functional architectures are built of functions only and are independent of the physical components that implement the functions. The functional architecture is more stable than a physical architecture that depends on the frequently changing technologies. The architecture principle to separate stable from unstable parts is covered in Chapter 9 about architecture patterns and principles.

Besides the functional architecture, we define and discuss further system architecture kinds. The base architecture that fixes the preset technologies and adjusts the scope for innovation, the logical architecture that specifies the technical concepts and principles, and the product architecture that finally specifies the concrete system. All three architecture kinds are physical architectures. The layered architecture is an additional aspect to these architecture kinds and is presented in Chapter 11.

Another additional aspect is the modeling of variants. Variability is increasingly important. The markets are no longer satisfied by commodity products. The market requests customized products that fit personal demands of the customers.

Additionally, global markets with different local environments and policies require different configurations of a system. Chapter 18 presents a model-based concept to specify different product configurations and gives a brief introduction to model-based product line engineering (MBPLE).

The architecture concepts are presented with a consistent example system. The "Virtual Tour" system (VT system) provides virtual visits by driving with camera-equipped robots through a real exhibition. The system is easy to understand and, at the same time, sufficiently complex to demonstrate the systems architecting concepts. The VT system is also part of a rescue and observation system to illustrate a system of systems and cyber-physical systems. The systems are introduced in Chapter 2.

The system architect who thinks that his job is to make a diagram and save it on a shared network drive will most probably fail. Same for the system architects who think they are the bosses of the development staff and can instruct the other engineers. It is neither an archaeological job nor a chief instructor job. Systems architecting is collaborative work that requires communication and soft skills. A basis for a good communication is a common language and media to transport the information. Chapter 8 covers the artifacts of the architecture documentations. In Chapter 19, we extend our scope to system of systems and architecture frameworks.

Typically, engineers are focused on the technology challenges of their job. As mentioned, communication and more general soft skills are getting more and more important capabilities. The engineering disciplines are growing together. For instance, that could be seen by the modern discipline mechatronic. And the worldwide humankind is growing together due to the internet other communication and transportation technologies. In consequence, an engineer has an increasing number of communication relationships. She is no longer successful when she only manages her technology tasks. It is also important to collaborate well with team members, stakeholders, communities, and so on. Chapter 23 gives an introduction about soft skills for engineers.

2

An Example: The Scalable Observation and Rescue System

We need an example system for the demonstration of various techniques to be presented in this book. The example shall be based on one single system with one single purpose, but extensible to be explored in scenarios involving the interaction of multiple systems for a purpose different from the one of the original single system.

Our single-purpose system is based on the very old idea of telepresence robots (e.g. [249]). The concrete system was inspired by an organization called "The Workers." They created a robot system that is called "After Dark" [36], because it is intended to be operated at night, when it is dark – and when almost any museum in town is closed. The system comprises robots that are driving through a closed museum. They carry a lamp to shed light and a camera to capture pictures. When sending the captured pictures to a remote user, the resulting offering is a virtual museum tour (VMT). The described system was demonstrated on 23 August 2014 [139]: After Dark's robots were driving through the gallery "Tate Britain", and people worldwide could watch the streamed camera images. A similar virtual museum tour offering based on a robot was started in The Mob Museum, Las Vegas, in 2016 [258], and the same technology was at least considered for several art museums [41].

Inspired by these systems, we present the "Virtual Museum Tour system" (VMT). Its central subsystem will be a robot as shown in Figure 2.1, intended for realizing a remote user's telepresence in a museum. The presented robot is based on some really existing prototype that was created many years ago during a leisure activity by two students (Erik Solda and Jesko G. Lamm) at Aachen university, Germany. To get back from this historic robot prototype to the example system considered here, please imagine the shown fictitious museum tour robot to be an industrial product with today's technologies onboard: It will use latest artificial intelligence (AI) to be able to navigate autonomously in a museum. But of course the system also comprises servers to control such robots, cloud services to offer

Model-Based System Architecture, Second Edition. Tim Weilkiens, Jesko G. Lamm, Stephan Roth, and Markus Walker.

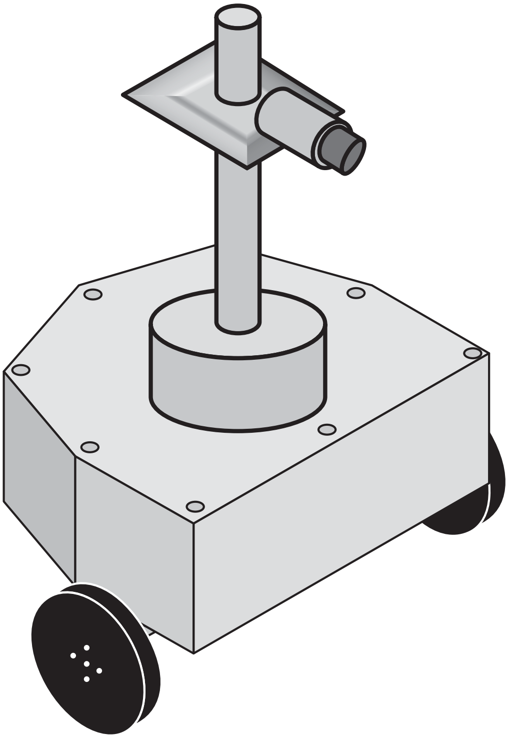


Figure 2.1 The museum tour robot.

onboarding to people worldwide, and apps for mobile devices to schedule virtual museum tours and watch the corresponding video streams.

A storyboard [152] in Figure 2.2 explains the system's main use case: Currently, John is controlling a museum robot to drive it through a museum of arts. He has to write a report about modern art as a homework for school, and he has not had time to go to the museum during its opening hours. John types "Andy Warhol" on his smartphone and the robot starts driving to the pop arts division of the museum. Once there, it stops in the middle of a room. John now selects a painting showing a soup can. The robot moves toward the painting and stops in front of it. The camera on the robot now transmits a picture of the painting to John's smartphone. A little notification box on the smartphone displays the title of the painting. John needs to analyze the artist's way of working in more detail. Via commands entered on his smartphone, he moves the camera down. Then, he zooms in on a particular area of the painting. Now he can see the necessary details via the video stream on his smartphone. This enables John to complete his homework for school.

Unlike the initially mentioned systems at Tate Britain and The Mob Museum, our own example system is purely fictitious and also the extensions to be presented