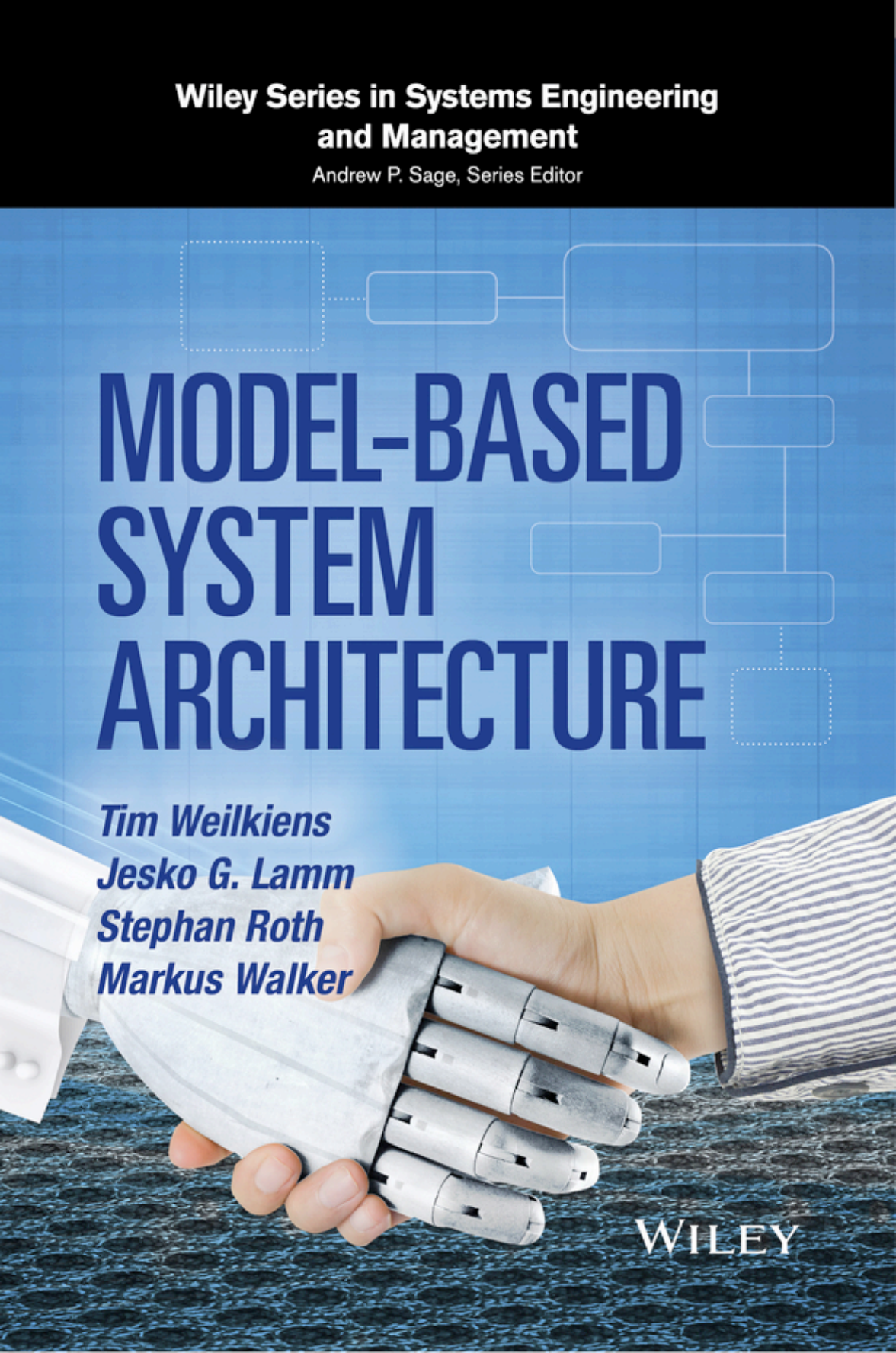


**Wiley Series in Systems Engineering  
and Management**

Andrew P. Sage, Series Editor

The background of the cover is a vibrant blue with a subtle grid pattern. In the upper right, there are several white-outlined diagrams representing system architecture, including a flowchart with rectangular boxes and a hierarchical tree structure. In the lower half, a human hand in a striped shirt sleeve is shaking a silver, segmented robotic hand. The overall theme is the integration of human and machine systems.

# **MODEL-BASED SYSTEM ARCHITECTURE**

*Tim Weilkiens*

*Jesko G. Lamm*

*Stephan Roth*

*Markus Walker*

**WILEY**



# **MODEL-BASED SYSTEM ARCHITECTURE**

---

**WILEY SERIES IN SYSTEMS ENGINEERING AND  
MANAGEMENT**

---

**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**

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

**WILEY**

Copyright © 2016 by John Wiley & Sons, Inc. All rights reserved

OMG SysML, OMG Systems Modeling Language, UML, Unified Modeling Language, Model Driven Architecture, MDA, Business Motivation Model, BMM and XMI are trademarks or registered trademarks of the Object Management Group, Inc. in the United States, the European Union and other countries.

Cameo is a registered trademark of No Magic, Inc.

FAS is a registered trademark of oose Innovative Informatik eG in Germany.

SYSMOD is a registered trademark of Tim Weilkiens in Germany.

V-Modell is a registered trademark of Bundesrepublik Deutschland in Germany.

TOGAF® is a trademark of The Open Group.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey Published simultaneously in Canada.

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, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

***Library of Congress Cataloging-in-Publication Data:***

Weilkiens, Tim.

Model-based system architecture / Tim Weilkiens, Jesko G. Lamm, Stephan Roth, Markus Walker.

pages cm. – (Wiley series in systems engineering and management)

Includes index.

ISBN 978-1-118-89364-7 (hardback)

1. System design. 2. Computer simulation. 3. SysML (Computer science) I. Lamm, Jesko G., 1976- II. Roth, Stephan, 1968- III. Walker, Markus, 1965- IV. Title.

TA168.W4338 2016

620'.0042–dc23

2015024774

Typeset in 11pt/13pt TimesTenLTStd by SPi Global, Chennai, India

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

1 2016

# Contents

---

<b>Foreword</b>	<b>xi</b>
<b>Preface</b>	<b>xv</b>
<b>About the Companion Website</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 An Example: The Virtual Museum Tour System</b>	<b>5</b>
<b>3 Better Products — The Value of Systems Architecting</b>	<b>9</b>
3.1 The Share of Systems Architecting in Making Better Products	9
3.2 The Benefits that can be Achieved	10
3.3 The Benefits that can be Communicated inside the Organization	14
3.4 The Beneficial Elements of Systems Architecting	15
3.5 Benefits of Model-Based Systems Architecting	16
<b>4 Definition of System Architecture</b>	<b>19</b>
4.1 What is Architecture? –Discussion of Some Existing Definitions	20
4.2 Modeling the Definitions of “System” and “System Architecture”	22

<b>5</b>	<b>Model-Based System Architecture</b>	<b>27</b>
<b>6</b>	<b>Architecture Description</b>	<b>35</b>
6.1	Why Spending Effort to Describe the Architecture?	35
6.2	The Architecture Description	37
6.3	How to Get an Architecture Description?	44
<b>7</b>	<b>Architecture Patterns and Principles</b>	<b>49</b>
7.1	The SYSMOD Zigzag Pattern	50
7.2	The Base Architecture	57
7.3	Cohesion and Coupling	61
7.4	Separation of Definition, Usage and Run-Time	63
7.5	Separate Stable from Unstable Parts	65
7.6	The Ideal System	65
7.7	View and Model	66
7.8	Diagram Layout	68
7.9	System Model Structure	69
7.10	Heuristics	71
<b>8</b>	<b>Requirements and Use Case Analysis</b>	<b>75</b>
8.1	Identify and Define Requirements	76
8.2	Specify the System Context	80
8.3	Identify Use Cases	82
8.4	Describe Use Case Flows	84
8.5	Model the Domain Knowledge	86
<b>9</b>	<b>Perspectives, Viewpoints and Views in System Architecture</b>	<b>89</b>
9.1	Overview	89
9.2	The Functional Perspective	91
9.3	The Physical Perspective	96
9.4	The Behavioral Perspective	100
9.5	The Layered Perspective	100
9.6	System Deployment Perspective	112



9.7	Other Perspectives	115
9.8	Relation to the System Context	117
9.9	Mapping Different Perspectives and Levels	120
9.10	Traceability	125
9.11	Perspectives and Views in Model-Based Systems Architecting	125
<b>10</b>	<b>Typical Architecture Stakeholders</b>	<b>131</b>
10.1	Overview	131
10.2	Requirements Engineering	133
10.3	Verification	135
10.4	Configuration Management	137
10.5	Engineering Disciplines	138
10.6	Project and Product Management	141
10.7	Development Roadmap Planners	145
10.8	Production and Distribution	148
10.9	Suppliers	149
10.10	Marketing and Brand Management	150
10.11	Management	152
<b>11</b>	<b>Roles</b>	<b>157</b>
11.1	Roles	157
11.2	The System Architect Role	158
11.3	System Architecture Teams	162
11.4	System Architecture Stakeholders	164
11.5	Recruiting System Architecture People	165
11.6	Talent Development for System Architects	167
<b>12</b>	<b>Processes</b>	<b>173</b>
12.1	The Systems Architecting Processes	173
12.2	Change and Configuration Management Processes	182
12.3	Other Processes Involving the System Architect	182

<b>13</b>	<b>Agile Approaches</b>	<b>183</b>
13.1	The History of Iterative-Incremental and Agile Development	184
13.2	System Architects in an Agile Environment	186
<b>14</b>	<b>The FAS Method</b>	<b>189</b>
14.1	Motivation	190
14.2	Functional Architectures for Systems	192
14.3	The FAS Method	195
14.4	FAS Heuristics	199
14.5	FAS with SysML	202
14.6	Modeling Tool Support	210
14.7	Mapping of a Functional Architecture to a Physical Architecture	215
14.8	Experiences with the FAS Method	218
14.9	FAS Workshops	219
14.10	Nonfunctional Requirements and the Functional Architecture	222
14.11	Completeness of the Functional Architecture	224
14.12	Functional Architectures and the Zigzag Pattern	227
<b>15</b>	<b>Product Lines &amp; Variants</b>	<b>231</b>
15.1	Definitions Variant Modeling	232
15.2	Variant Modeling with SysML	233
15.3	Other Variant Modeling Techniques	239
<b>16</b>	<b>Architecture Frameworks</b>	<b>243</b>
16.1	Enterprise Architectures	244
16.2	System of Systems (SoS)	246
16.3	An Overview of Architecture Frameworks	249
16.4	The UPDM Standard	261
16.5	What to do when we Come in Touch with Architecture Frameworks	262
16.6	Conclusion	263

<b>17</b>	<b>Cross-Cutting Concerns</b>	<b>265</b>
17.1	The Game-Winning Nonfunctional Aspects	265
17.2	Human System Interaction and Human Factors Engineering	266
17.3	Risk Management	267
17.4	Trade Studies	268
17.5	Budgets	269
<b>18</b>	<b>Architecture Assessment</b>	<b>271</b>
<b>19</b>	<b>Making it Work in the Organization</b>	<b>277</b>
19.1	Overview	277
19.2	Organizational Structure for Systems Architecting	278
19.3	Recipes from the Authors' Experience	283
<b>20</b>	<b>Soft Skills</b>	<b>291</b>
20.1	It's all about Communication	292
20.2	Personality Types	303
20.3	Intercultural Collaboration Skills	307
<b>21</b>	<b>Outlook: The World after Product Line Engineering</b>	<b>311</b>
<b>A</b>	<b>OMG SysML</b>	<b>315</b>
A.1	Diagram and Model	316
A.2	Structure Diagrams	318
A.3	Behavior Diagrams	329
A.4	Requirements Diagram	337
A.5	Extension of SysML with Profiles	340
A.6	Architecture of the Language	341
<b>B</b>	<b>The V-Model</b>	<b>343</b>
B.1	A Brief History of the V-Model or the Systems Engineering VEE	343
B.2	A Handy Illustration but No Comprehensive Process Description	345
B.3	Critical Considerations	348

B.4	Reading Instruction for a Modern Systems Engineering VEE	351
	<b>Bibliography</b>	<b>353</b>
	<b>Index</b>	<b>363</b>

# 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 ten 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. The use of disjoint models and simulation in systems engineering is also not 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 the 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.

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 on 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 architect for his daily challenges. The scope of the book is a model-based environment, either it is 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 system architecting, which we see as a required backbone for excellent systems 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 to only choose the approaches that are most suitable for your daily work and disregard the others, until you are in a situation where they turn out to be useful. 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 that we find easy to apply in daily work and 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 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 museum visits as an example system.

We like to write texts using gender-fair language. On the other hand, we avoid to clutter 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 wherever it is appropriate.

We like to thank the “FAS” and “MkS” working groups of GfSE, the German 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 works, who have influenced our way of thinking, helped us with foreign languages in both reading and writing or recommended literature that is 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 hear 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, his assistants Alex Castro and Kathleen Pagliaro, and Bhargavi Natarajan for their support.

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

Contributor  
MATTHIAS DÄNZER,  
Bernafon AG  
February 2015

# About the Companion Website

---

This book is accompanied by a companion website:

[www.mbse-architecture.com](http://www.mbse-architecture.com)

The website includes:

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



## Chapter 1

---

# Introduction

---

Model-based system architecture (MBSA) combines the two key technologies: model-based and system architecting. Both are major parts of the future state of systems engineering [57].

Many systems result from an evolutionary development. They are driven by their parts and do not emerge from the architecture. The parts could be anything that in combination are assembled to a man-made technical system. System architecture is exhibited 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 [20]. We understand system architecture is more holistic and also consider systems without any software. Although systems without any software that are handled with systems engineering processes and model-based system architecture concepts like described in this book are very rare, a system architecture is always present. In today's and future systems engineering, it is crucial to apply explicit system architecting for the success of the system project [57]. Chapter 4 defines the term *system architecture* within its context.

Studies clearly show that system architecting is critical for the performance and success of the system [34]. This is particularly evident for projects that require significant architectural work or rework. Due to more and more dynamic and complex markets and environments,

*Model-Based System Architecture*, First Edition.

Tim Weilkiens, Jesko G. Lamm, Stephan Roth, and Markus Walker.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.

the system architecture must more and more withstand the changing requirements and requests for radical changes. Chapter 3 lists the benefits of system architecting.

System architecture is about establishing solutions that are 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 tool to be capable of developing complex systems on time and in a feasible quality. Chapter 5 defines the term *model* and MBSA and discusses the 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 semantics, which you find in a modeling language. We use the international standard Systems Modeling Language (SysML) as language for the system requirements and architecture models. Appendix A gives an overview about SysML. Although we extensively use SysML in this book, our methods and concepts are independent on SysML and could also be implemented by other modeling languages.

The system architect is the one in charge of shaping the system architecture. This is a big responsibility and a big challenge. The organizations dependent on the system should carefully select the 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 19 describes how system architecting could be embedded in an organization and Chapter 10 discusses the interfaces to the stakeholders of system architecting. In particular Chapter 8 introduces the adjacent discipline requirements engineering that closely collaborates with the system architecting. The SYSMOD zigzag pattern presented in Chapter 7 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 FAS method.

Chapter 14 is a comprehensive presentation of the Functional Architectures for Systems (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 steadily changing technologies. The architecture principle to separate stable from unstable parts is covered in Chapter 7 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 orthogonal aspect to these architecture kinds and is presented in Chapter 9.

Another orthogonal 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 to personal demands of the customers. In addition, global markets with different local environments and policies require different configurations of a system. Chapter 15 presents a model-based concept to specify different product configurations.

The architecture concepts are presented with a consistent example system. The “Virtual Museum Tour” system provides virtual visits by driving with camera-equipped robots through a real museum. The system is easy to understand and at the same time is sufficiently complex to demonstrate the system architecting concepts. The system is 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 archeological job nor a chief instructor job. System architecting is a collaborative work that requires communication and soft skills. The basics for a good communication is a common language and media to transport the information. Chapter 6 covers the artifacts of the architecture documentations. In Chapter 16, we extend our scope to system of systems and architecture frameworks.

Typically, engineers are focused on the technology challenges of their job. Nowadays, 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 mankind is growing together due to the Internet, other communication, and transportation technologies.

#### **4**    *INTRODUCTION*

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 20 gives an introduction about soft skills for engineers.

## Chapter 2

---

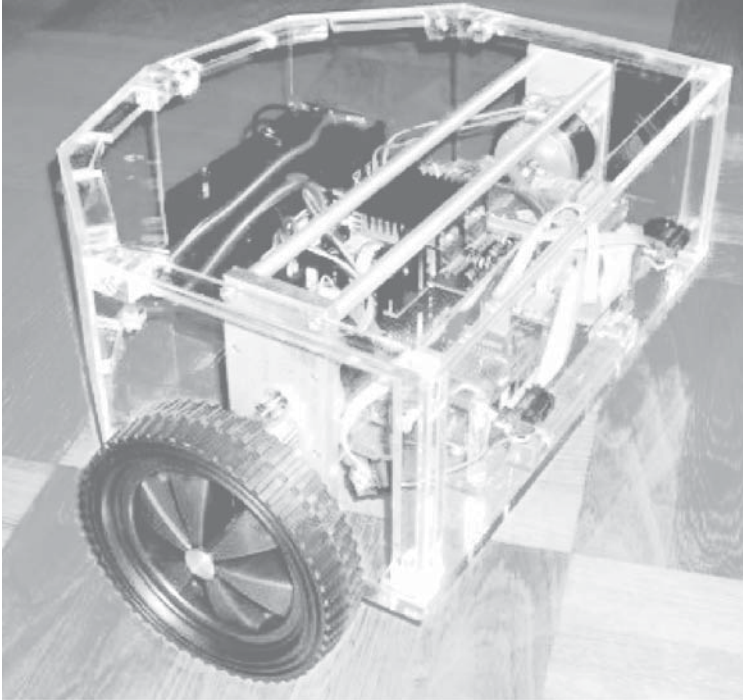
# An Example: The Virtual Museum Tour System

---

We need an example system for the demonstration of various techniques to be presented in this book. The example of our choice is based on robotics work that one of us (Jesko G. Lamm) started doing as a leisure activity during his time at university, together with his fellow student Erik Solda who had the initial idea. During the course of this work, different robots were built. One of them is shown in Figure 2.1. It is the one that inspired the authors of this book in developing a fictitious example system to be presented in the following.

For this book, we thought of a robot according to Figure 2.2. The robot would drive through a museum. A live video stream from a camera on the robot would be broadcasted. People with access to the live video stream could thus see the exposition, even when the museum is closed.

When we had completed the first chapters of this book based on this example system, we noticed that we were not the only ones who had this idea: an organization called “The Workers” actually had this idea prior to us and invented a museum robot system that is called “After Dark”, because it is intended to run during the night when the museum is closed and dark. The Workers’ idea of the After Dark system won the

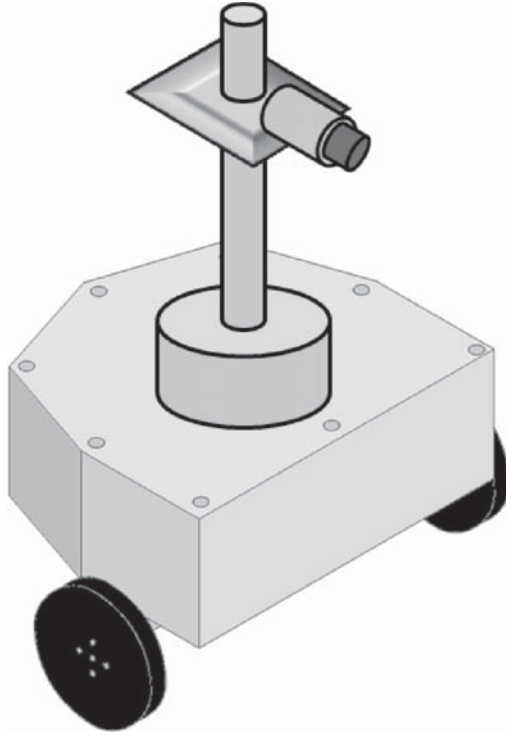


**Figure 2.1.** One of the robots that was built during the robotics work by J.G. Lamm and E. Solda. © 2007, 2014 Jesko G. Lamm, reproduced with permission.

so-called “IK Prize”—including a budget that allowed them to actually build the system and offer the first virtual tour through the gallery Tate Britain on August 13, 2014 [74].

The After Dark system came to our attention quite late during the process of writing this book. Therefore, the system we use as an example here is mainly based on our own imagination and may thus be quite different from the After Dark system, for example, regarding its architecture. Still, the After Dark system has become an important source of inspiration after Tate’s press release about it [17] had been discovered by one of the authors.

In our fictitious system according to Figure 2.2, one user can log in for controlling the robot via a web application on a computer or a hand-held device, like a smart phone. Multiple users can watch the video stream from the camera via a web application. The camera can zoom and move up and down. Roughly speaking, the system-of-interest is composed



**Figure 2.2.** The fictitious museum robot with the purpose of enabling virtual museum tours.

of the robot itself together with the infrastructure and remote control software that is needed to operate it. We call this system the “Virtual Museum Tour system (VMT)”. Chapter 9 will provide example architecture descriptions that provide a more precise definition of the system.

A little story may help understanding the capabilities of the example system: 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 smart phone 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 smart phone. A little notification box on the smart phone displays the title of the painting. John needs to analyze the artist’s way

of working in more detail. Via commands entered on his smart phone, 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 smart phone. This enables John to complete his homework for school.