



Wolfgang Mahnke  
Stefan-Helmut Leitner  
Matthias Damm

# OPC Unified Architecture

 Springer

# OPC Unified Architecture

Wolfgang Mahnke • Stefan-Helmut Leitner •  
Matthias Damm

# OPC Unified Architecture

 Springer

Wolfgang Mahnke  
Stefan-Helmut Leitner  
ABB Forschungszentrum  
Ladenburg  
Wallstadter Str. 59  
68526 Ladenburg  
Germany  
wolfgang.mahnke@de.abb.com  
stefan.leitner@de.abb.com

Matthias Damm  
ascolab GmbH  
Automation Systems  
Communication Lab.  
Am Weichselgarten 7  
91058 Erlangen  
Germany

ISBN 978-3-540-68898-3

e-ISBN 978-3-540-68899-0

DOI 10.1007/978-3-540-68899-0

Library of Congress Control Number: 2008941149

© 2009 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permissions for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

*Cover design:* KuenkelLopka GmbH

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

# Foreword by Tom Burke

The OPC Foundation is very honored to endorse this superior book and the excellent work that the authors have put together. This book provides a solid framework of understanding about the OPC Foundation specifications and technology from the beginning of OPC up through and including the most important OPC interoperability standard, that being the OPC Unified Architecture.

The authors of this book, I have been pleased to know for many years, clearly are the architectural and development leaders who have enabled OPC to be so widely successful. The readers of this book are fortunate to be able to learn from the experts who actually developed the OPC Foundation Unified Architecture specifications and technology.

The OPC Unified Architecture was developed by over 30 companies over approximately 5 years. Complete reference implementations and technology were developed to validate the specifications and prove the technical feasibility. The purpose of the OPC Unified Architecture was to enable a platform-independent interoperability standard for moving data/information between the factory floor and the enterprise. During the course of the development it was obvious that the OPC Unified Architecture was well positioned to expand beyond industrial automation. OPC has expanded into areas of building automation, security, home automation, power generation, packaging, and petrochemicals. Because of the highly scalable architecture of OPC UA, it is also well-positioned for deployment in intelligent embedded devices.

OPC UA is a collaborative effort with other standards organizations as well. OPC UA is also built on the premise of do not reinvent technology that already exists. OPC pulls all the pieces together as necessary for true secure reliable interoperability.

This book provides you a solid foundation to learn everything you could ever want to know about developing world-class products for multi-vendor interoperability based on OPC UA.

The OPC Foundation is proud to recognize the achievements and quality work that the authors have put together in developing and assembling this book. I encourage you to read this book multiple times and use it as a constant reference as you develop or use OPC-based products in your respective domain. You as a reader are very fortunate to have obtained this book.

I encourage you to constantly refer and take advantage of this book for all your OPC needs.

Tom Burke  
President and Executive Director OPC Foundation

# Foreword by Jim Luth

More than 5 years in the making, OPC Unified Architecture represents a revolutionary step forward for vendors wishing to write software that interoperates with others. A group of dedicated volunteer members of the OPC Foundation contributed countless hours to complete this ambitious endeavor. By combining the tried and true functionality of the previous generation of OPC Interfaces along with the latest advances in computer science (e.g., Object Oriented Programming, Service Oriented Architecture, the Semantic Web, Network Model Databases), OPC UA represents a generic framework for exposing and consuming data and metadata of any complexity.

In the mid-nineties, the OPC Foundation published its first Microsoft COM specification for Data Access (DA) that specified how a server would expose a simple hierarchical organization of items (tags) that could be read, written, and subscribed to by conformant clients. The Foundation quickly followed up with additional popular specifications for different types of data, in particular, Alarms & Events and Historical Data. With the invention of XML Web Services and the promise of vendor neutral communication, the Foundation created XML-DA, a platform independent Web Service interface for Data Access with similar functionality as the original COM version. Unfortunately, the performance of the Web Service version was orders of magnitude slower than the COM version, so XML-DA could not be viewed as a newer and better replacement for the platform specific COM version. The OPC Unified Architecture was born out of the desire to create a true replacement for all of the existing COM-based specifications without losing any features (or performance) in the process. Here are some of the design goals of OPC UA:

- Support a wider range of applications that use complex instead of simple data (MES, ERP, Asset Management...).
- Allow Data Access, Alarms & Events, and Historical data to be exposed using a single set of generic, data-agnostic Services.
- Allow the nodes in the address space to be connected in hierarchies and non-hierarchical “meshes.”
- Future-proof the specifications by making them abstract and not dependent on existing communication technologies.
- Specify concrete data serializations and protocol mappings using accepted internet standards (Web Services, XML, HTTP, TCP ...)
- Allow rich metadata to be exposed (the same way the data itself is), so that generic clients can interpret data without a priori knowledge.

All these goals and more have been realized in OPC UA. We now have implementable specifications, communication stacks, and SDKs in multiple programming

languages and higher level third-party toolkits. We now invite the rest of the world to go forth and create software applications that interoperate at the highest possible semantic levels using OPC UA.

With the reach-for-the-stars design goals and the huge scope of OPC UA, the specifications, currently comprised of 13 Parts and climbing, while terse, dense, and exact, are not the easiest way for developers and architects to approach OPC UA. I like to use the corollary to SQL. There are thousands of software developers skilled at developing SQL applications, yet most have never read the SQL specifications. They learned SQL by reading books, studying vendor documentation, taking classes, etc. This book, written by key authors and contributors to the OPC UA specification and coding effort, represents the best way to learn and use OPC UA in your programming and design tasks.

Jim Luth  
Technical Director OPC Foundation

# Preface

## Motivation for This Book

The OPC Foundation provides specifications for data exchange in industrial automation. There is a long history of COM/DCOM-based specifications, most prominent OPC Data Access (DA), OPC Alarms and Events (A&E), and OPC Historical Data Access (HDA), which are widely accepted in the industry and implemented by almost every system targeting industrial automation.

Now the OPC Foundation has released a new generation of OPC specifications called OPC Unified Architecture (OPC UA). With OPC UA, the OPC Foundation fulfills a technology shift from the retiring COM/DCOM technology to a service-oriented architecture providing data in a platform-independent manner via Web Services or its own optimized TCP-based protocol. OPC UA unifies the previous specifications into one single address space capable of dealing with current data, alarms and events and the history of current data as well as the event history. A remarkable enhancement of OPC UA is the Address Space Model by which vendors can expose a rich and extensible information model using object-oriented techniques. OPC UA scales well from intelligent devices, controllers, DCS, and SCADA systems up to MES and ERP systems. It also scales well in its ability to provide information; on the lower end, a model similar to Classic OPC can be used, providing only base information, while at the upper end, highly sophisticated models can be described, providing a large amount of metadata including complex type hierarchies.

There is a high interest in the advanced modeling capabilities in many domains and there are already initiatives to standardize information models based on OPC UA. Examples of these activities are FDI where a common field device description is targeted and common activities with MIMOSA (Maintenance Information – ERP and above), S95 (Production Information – MES), and PLCopen (Industrial Control).

The OPC UA specification currently consists of 13 parts and therefore 13 documents, some specifying the base technology and others defining specific information models. An example is a model describing how to provide process automation-specific alarm information. All in all, there are over 700 pages of specification, written to be accurate and complete. The specification primarily defines how to do things and to a lesser extent explains why it was designed that way. This is the way specifications have to be written to be widely applied by many developers and to guarantee interoperability between different applications. As a result, the OPC UA specification is hard to read for someone new to OPC UA. This is also true of other specifications such as SQL or UML.

With this book, we want to fill this gap and provide an easy to understand introduction to OPC UA. We will not provide the same level of detail as the specification, but rather introduce and explain the main concepts of OPC UA. We will give



guidelines that help you in determining the best alternative among different concepts for your use cases and requirements. We will also target relevant topics that are not directly addressed in the specification but are needed to apply OPC UA.

## Who Should Read This Book?

If you are interested in the OPC Unified Architecture – and that is probably the reason why you are reading this text – you should read this book. It is written by the editors of the key parts of the OPC UA specification, and they will explain to you what is behind the acronym OPC UA.

We had a broad audience in mind when writing this book, including people with the following tasks:

- Judging whether OPC UA should be applied in their applications (decision makers)
- Applying OPC UA in their client or server applications (software architects, engineers, and developers)
- Using applications based on OPC UA (administrators and engineers, e.g., responsible for configuring a process control system; not end users such as operators of a process control system).

This book will introduce the communication and information modeling concepts of OPC UA. It will explain how to define your model and how to access the data. You will learn how redundancy, security, and more are addressed in OPC UA and how well it performs compared to Classic OPC. However, you will not find any code examples. When you implement your OPC UA application, it is expected that you will use an SDK and you should look into the documentation of such an SDK for coding examples. This book explains how the mechanisms behind any OPC UA SDK – the OPC UA Services – work and how to model your information in OPC UA. It also explains the information you can expect when accessing an OPC UA server.

You do not have to be familiar with Classic OPC to read this book. You should have a basic understanding of object-oriented concepts in order to understand the information modeling. Some basic knowledge of software architecture is needed to understand the underlying architecture of OPC UA.

## Outline

Chapter 1 gives a short introduction into Classic OPC before starting with the motivation for OPC UA and giving a short overview of OPC UA.

The next three chapters focus on information modeling, that is, how data can be represented using OPC UA. Chapter 2 introduces the modeling concepts. We start with the fundamental concepts to provide data, and later introduce more sophisticated constructs such as type hierarchies. Chapter 3 introduces a real-life example of how to model information in OPC UA and then generalizes modeling by explaining some best practices. Standard information models are introduced in Chap. 4. Starting by explaining what an information model is and how it can be specified as well as how OPC UA deals with information models. Continuing, the base OPC UA information model is introduced followed by more specialized information model extensions of the OPC UA specification. Finally, we will look at the current state of additional information model standards provided by other organizations.

The next two chapters focus on the way how to access the information modeled in OPC UA. In Chap. 5, the abstract Services are described, which are used to access or manipulate data. In Chap. 6, the mapping of those Services to concrete technology is introduced. It is described how data is serialized, how messages are secured, and what transport protocol is used.

In Chap. 7, the security considerations of OPC UA are discussed. This includes the theoretical thoughts behind the security model of OPC UA as well as the practical implications for developers and administrators of OPC UA applications.

Chapter 8 explains the application architecture of OPC UA. Here, the different components needed to implement OPC UA are introduced.

In Chap. 9, the system architecture of OPC UA is described. Included are descriptions of how you can deploy and configure OPC UA applications in your systems, and how to handle redundancy, aggregation of servers, etc.

In the next chapters, we focus on migration of existing applications to OPC UA. In Chap. 10 we explain how concepts of Classic OPC map to the concepts of OPC UA. This chapter is especially useful for readers having a deep knowledge of Classic OPC. Chapter 11 provides strategies of how to migrate your Classic OPC applications to OPC UA and how components, provided by the OPC Foundation, can help you in this effort.

OPC UA specifies a large number of features, but not every application will make use of all of them. OPC UA provides profiles to deal with this fact. Profiles specify a subset of features a product ensures to support. Applications exchange these profiles to know what they can expect from the other application. Details on profiles and how they are organized are described in Chap. 12.

Performance is a critical factor in the scenarios where Classic OPC is typically used today. In Chap. 13, performance considerations of OPC UA are given. This includes a comparison of OPC UA performance to Classic OPC.

We close with Chap. 14 where we summarize OPC UA and discuss the complexity of OPC UA, pointing out that it is simple in most cases and explain why some parts must have some complexity. We also provide an outlook of what we expect to happen in the near future regarding OPC UA.

In addition, some Appendices provide a quick reference when you need to find some details of OPC UA.

# About the Authors

## **Dr. Wolfgang Mahnke**

Wolfgang Mahnke works at the ABB Corporate Research Center in Germany in the field of Industrial Software Technology. In recent years, he has been the project leader of several projects related to the OPC Unified Architecture. Those projects target the specification of OPC UA, the implementation of the infrastructure provided by the OPC Foundation as well as the application of OPC UA inside ABB, for example, in ABB's major DCS 800xA. He is editor of the Address Space Model and the Information Model parts of the OPC UA specification and has conducted several OPC UA training sessions and given several presentations over the last years.

Wolfgang holds a Diploma in Computer Science from the University of Stuttgart. During his work at the University of Kaiserslautern he gained his Ph.D. in the area of Databases and Information Systems.



## **Stefan-Helmut Leitner**

Stefan-Helmut Leitner works at the ABB Corporate Research Center in Germany in the area of Industrial Software Technologies. He has been involved in various research and development topics regarding OPC Unified Architecture inside and outside ABB, such as the development of the ANSI-C protocol stack, certificate management for OPC UA, and held several trainings and presentations. In addition, he has the editing responsibility for the Security Model part of the OPC UA specification.

Stefan-Helmut holds a Diploma in Information Technology from the University of Corporate Education Mannheim.



## **Matthias Damm**

Matthias Damm is Executive Director and founder of ascolab, where he is responsible for OPC consulting and certification. Since the last 10 years, he has been actively involved in OPC work especially in the areas of compliance testing and OPC Unified Architecture.

Before founding ascolab, Matthias worked as manager of the OPC competence centre in the Siemens division Industrial Services and Solutions.

Matthias is manager of the OPC Foundation Certification Test Lab at ascolab and his team is responsible for the development and maintenance of the OPC Foundation Compliance Test Tools since several years.

He is editor of the Service part of the OPC UA specification and has conducted OPC UA training sessions and given several presentations over the last years. He was involved in the design of the portable ANSI C UA Stack developed by ascolab that was donated to the OPC Foundation. He was also responsible for the design and development of the C++ UA Server SDK used by many early OPC UA products, which is distributed by Unified Automation GmbH.

Matthias has a Dipl. Ing. (FH) degree in Electrical Engineering from University of applied Science in Schweinfurt/Germany.



# Acknowledgements

Writing a book is a time consuming task that requires many extra after-office hours and weekend hours. First of all, we thank our families and friends for their patience and support. Helmut and Wolfgang thank Martin Naedele from ABB for sponsoring some of our time at the office for this book project. Matthias thank his co-executives at ascolab, Gerhard Gappmeier and Uwe Steinkrauß.

The topic of this book – OPC Unified Architecture – was created by the OPC UA working group of the OPC Foundation. We gained our knowledge and thereby the foundation for writing this book by being part of this working group. We always enjoyed the positive atmosphere of the group focusing on technology issues and not on politics. A lot of people have participated in the working group meetings, some from the beginning to the present and some only for a while. We thank all of them for the excellent cooperation. We specially thank Jim Luth of the OPC Foundation as the leader of the working group, and those who were involved in so many discussions: Randy Armstrong and Tom Burke (OPC Foundation), Jeff Harding and Paul Hunkar (ABB), Karl-Heinz Deiretsbacher (Siemens), Lee Neitzel (Emerson), Ayana Craven (OSIsoft), Erik Murphy (Matrikon), Christian Zugfil (ascolab), Jörg Allmendinger (Allmendinger), and Betsy Hawkinson (Honeywell), to name a few.

Several reviewers helped improve the quality of this book. We thank Randy Armstrong (OPC Foundation), Karl-Heinz Deiretsbacher (Siemens), Jens Doppelhamer (ABB), Gerhard Gappmeier (ascolab), Jeff Harding (ABB), Paul Hunkar (ABB), Emanuel Kolb (ABB), Heiko Koziol (ABB), Claude Lafond (ABB), Jim Luth (OPC Foundation), Uwe Steinkrauß (ascolab), and Roland Weiss (ABB) for their assistance.

Thanks also to our publisher for their support, especially to Dorothea Glaunsinger, Hermann Engesser, and the native speakers who greatly improved our English.

Finally, we thank all those people we have had discussions with about OPC UA at DevCons, workshops, trainings, and other occasions. Those discussions helped us a great deal identifying key problems with understanding OPC UA and gave us the motivation to write this book.

To the readers of this book: We encourage you to contact us if you find errors or unclear statements in this book or if you have suggestions for improvement. We will post corrections and additional information on [www.opcuabook.com](http://www.opcuabook.com).

October 2008

Wolfgang Mahnke ([wolfgang.mahnke@de.abb.com](mailto:wolfgang.mahnke@de.abb.com))

Stefan-Helmut Leitner ([stefan.leitner@de.abb.com](mailto:stefan.leitner@de.abb.com))

Matthias Damm ([matthias.damm@ascolab.com](mailto:matthias.damm@ascolab.com))

# Table of Contents

<b>Foreword by Tom Burke</b> .....	<b>v</b>
<b>Foreword by Jim Luth</b> .....	<b>vii</b>
<b>Preface</b> .....	<b>ix</b>
<b>About the Authors</b> .....	<b>xiii</b>
<b>Acknowledgements</b> .....	<b>xv</b>
<b>1 Introduction</b> .....	<b>1</b>
1.1 OPC Foundation .....	1
1.2 Classic OPC .....	3
1.2.1 OPC Data Access .....	4
1.2.2 OPC Alarm and Events .....	5
1.2.3 OPC Historical Data Access .....	6
1.2.4 Other OPC Interface Standards .....	6
1.2.5 OPC XML-DA .....	7
1.3 Motivation for OPC UA .....	8
1.4 OPC UA Overview .....	10
1.5 OPC UA Specifications .....	11
1.6 OPC UA Software Layers .....	13
1.7 Evolution Not Revolution .....	15
1.8 Summary .....	16
1.8.1 Key Messages .....	16
1.8.2 Where to Find More Information? .....	17
1.8.3 What's Next? .....	17
<b>2 Information Modeling: Concepts</b> .....	<b>19</b>
2.1 Why Information Modeling? .....	19
2.2 Nodes and References .....	22
2.3 ReferenceTypes .....	25
2.4 Objects, Variables and Methods .....	30
2.5 Types for Objects and Variables .....	36
2.5.1 Simple ObjectTypes .....	37
2.5.2 Simple VariableTypes .....	39
2.5.3 Complex ObjectTypes .....	42
2.5.4 InstanceDeclarations .....	45
2.5.5 Complex VariableTypes .....	48
2.5.6 ModellingRules .....	48
2.5.7 Subtyping of ComplexTypes .....	54
2.6 Data Variables and Properties .....	58

2.7	ModelParent for Objects, Variables and Methods .....	60
2.8	DataTypes .....	61
2.8.1	DataType NodeClass .....	62
2.8.2	Built-in and Simple DataTypes .....	63
2.8.3	Enumeration DataTypes .....	63
2.8.4	Structured DataTypes .....	64
2.8.5	Specific Built-in DataTypes .....	68
2.8.6	Summary on DataTypes .....	71
2.9	Views .....	71
2.10	Events .....	74
2.11	Historical Access .....	77
2.11.1	Historical Data .....	77
2.11.2	Historical Events .....	78
2.11.3	Historical Address Space .....	79
2.12	Address Space Model and Information Models .....	81
2.13	Summary .....	83
2.13.1	Key Messages .....	83
2.13.2	Where to Find More Information? .....	84
2.13.3	What's Next? .....	84
<b>3</b>	<b>Information Modeling: Example and Best Practice .....</b>	<b>85</b>
3.1	Overview .....	85
3.2	Example .....	85
3.2.1	Application Scenario .....	85
3.2.2	Simple Scenario – Similar to Classic OPC .....	87
3.2.3	Advanced Scenario – Providing Full Power of OPC UA .....	88
3.3	Best Practices .....	94
3.3.1	Structuring with Objects, ReferenceTypes and Views .....	95
3.3.2	TypeDefinitions (ObjectTypes and VariableTypes) .....	96
3.3.3	Providing Complex Data Structures .....	100
3.3.4	Providing User-Defined DataTypes .....	102
3.3.5	Properties .....	102
3.3.6	Methods .....	103
3.3.7	ModellingRules .....	103
3.3.8	Proxy Objects (Properties on References) .....	104
3.4	Summary .....	105
3.4.1	Key Messages .....	105
3.4.2	Where to Find More Information? .....	105
3.4.3	What's Next? .....	106
<b>4</b>	<b>Standard Information Models .....</b>	<b>107</b>
4.1	Overview .....	107
4.2	Handling Information Models .....	107
4.2.1	What is Specified by an Information Model? .....	107
4.2.2	How is an Information Model Specified? .....	111