

Table of Contents

Rainer Drath

1 About this book — 1

- 1.1 Introduction — 1
- 1.2 Reading guidance — 1
- 1.3 About AutomationML — 2
- 1.4 Related AutomationML specifications — 3
 - 1.4.1 Overview — 3
 - 1.4.2 AutomationML IEC 62714 Series — 3
 - 1.4.3 Whitepapers — 5
 - 1.4.4 Application Recommendations — 5
 - 1.4.5 Best practice recommendations — 5
 - 1.4.6 Availability — 6

Part I: AutomationML Development Support — 7

Rainer Drath

2 Overview of Part I — 9

Ronald Rosendahl, Ender Yemenicioglu

3 Software development with AutomationML — 11

- 3.1 Introduction — 11
- 3.2 Layered architecture — 11
 - 3.2.1 Layer 0 — 12
 - 3.2.2 Layer 1 — 13
 - 3.2.3 Layer 2 — 14
 - 3.2.4 Layer 3 — 15
- 3.3 Implementation Methods — 16
 - 3.3.1 Simple API for XML (SAX) — 16
 - 3.3.2 Document Object Model (DOM) — 16
 - 3.3.3 Serialization to XML — 17
 - 3.3.4 Implementation — 17
 - 3.3.5 Use cases — 18
- 3.4 Java Engine — 19
 - 3.4.1 Java Architecture for XML Binding (JAXB) — 19
 - 3.4.2 EclipseLink – JAXB model compiler — 20
 - 3.4.3 EclipseLink – persistence API — 22
 - 3.4.4 AutomationML Object Modelling — 24
- 3.5 AutomationML Engine — 28
 - 3.5.1 AML.Engine (C#), basic architecture and design — 28
 - 3.5.2 AML.Engine Tutorial — 30
- 3.6 AutomationML Editor Plugin — 35
 - 3.6.1 Getting Started — 35
 - 3.6.2 Plugin implementation — 36
- 3.7 Comparison and Conclusion — 38
- 3.8 About the Authors — 40

Josef Prinz, Ender Yemenicioglu, Rainer Drath

4 AutomationML Export and Import Data Interfaces — 41

- 4.1 Introduction — 41
- 4.2 General import/export concept — 41
 - 4.2.1 Overview — 41
 - 4.2.2 General architecture — 42
 - 4.2.3 Semantics during export and import — 44
 - 4.2.4 General import concept — 45
- 4.3 Implementation concepts for the transformation layer of an export interface — 46
- 4.4 Implementation concepts for the transformation layer of an import interface — 50
- 4.5 Definition of SystemUnitClasses for externalization concepts — 53
- 4.6 Identification concepts — 55
 - 4.6.1 Labelling for exporters — 55
 - 4.6.2 Object identification — 56
- 4.7 Interface Implementation examples — 58
 - 4.7.1 General Procedures — 58
 - 4.7.2 AutomationML Export with internalized generation — 62
 - 4.7.3 AutomationML Export with externalized generation — 66
 - 4.7.4 AutomationML Import with internalized generation — 70
- 4.8 Conclusion — 72
- 4.9 About the Authors — 73

Jan-Wilhelm Blume, Ljiljana Stojanovic

5 The AMLTestCenter — 75

- 5.1 Introduction and motivation for the AMLTestCenter — 75
- 5.2 Verification of general AML conformity with the AMLTestCenter — 77
 - 5.2.1 General concept — 77
 - 5.2.2 About consistency rules — 78
 - 5.2.3 User interface and application — 79
 - 5.2.4 Fields of improvement — 80
- 5.3 AMLTestCenter for domain specific rules — 80
 - 5.3.1 Requirements/Needs for a declarative approach — 80
 - 5.3.2 New concept -extension with the “engine” for rule execution — 82
 - 5.3.3 AAS rules example — 83
 - 5.3.4 Implementation details — 84
- 5.4 Conclusion and next steps — 85
- 5.5 About the Authors — 86

References for Part I — 87

Part II: The Industrial Cookbook — 91

Rainer Drath

6 Overview of Part II — 93

Rainer Drath, Thomas Tauchnitz

7 AML domain model for VDI 3697-1: Data exchange between CAE and PCS — 95

7.1	Introduction and Motivation —	95
7.2	About Process Control Equipment (PCE) —	97
7.3	GMA-Technical Committee 6.16 —	98
7.4	Architecture overview of a PCE request —	99
7.5	The AML Class model —	101
7.5.1	Overview —	101
7.5.2	Class model of the NE150SUCLibrary —	102
7.5.3	Data model of the NE150InterfaceLibrary —	104
7.5.4	Data model of the NE150PlantHierarchyRoleClassLib —	105
7.6	Namur Container modelling guideline —	106
7.6.1	Preparation —	106
7.6.2	Steps how to model a PCE request with AML —	107
7.7	Typical modelling guideline —	109
7.7.1	About Typicalals —	109
7.7.2	How to model typicalals —	109
7.8	Experiences in the practical implementation —	111
7.9	Discussion and Conclusion —	111
7.10	Acknowledgements —	112
7.11	About the Authors —	112

Andreas Schüller

8 AML domain model for VDI 3697-2: Data Exchange between CAE systems — 113

8.1	Introduction and Motivation —	113
8.2	Data model —	114
8.3	AutomationML library NE159SUCLibrary —	116
8.4	AML modelling guideline —	118
8.4.1	Application example according to NE 159 —	118
8.4.2	Step 1: Preparation —	119
8.4.3	Step 2: Modelling the plant hierarchy —	120
8.4.4	Step 3: Modelling the PCE request —	122
8.5	Discussion and Conclusion —	122
8.6	Acknowledgements —	123
8.7	About the Author —	123

Mario Hoernicke, Katharina Stark, Leon Urbas

9 MTP – Automation Engineering of Modular Process Plants — 125

9.1	Introduction —	125
9.1.1	Overview —	125
9.1.2	About modules, functionality and motivation —	126
9.1.3	State-of-the-Art —	127
9.1.4	Automation engineering and architecture of modular plants —	128
9.1.5	State-based control and service-oriented architecture —	130
9.1.6	Standardization —	130
9.1.7	Scope —	131
9.2	MTP and AutomationML —	131
9.3	Data model – The Module Type Package (MTP) —	132
9.4	AutomationML classes —	135
9.4.1	Basic libraries —	137
9.4.2	Data object libraries —	139

9.4.3	HMI libraries —	141
9.4.4	Service libraries —	142
9.4.5	Alarm libraries —	144
9.5	Modelling guide based on an application example —	145
9.5.1	How to model the Manifest —	145
9.5.2	How to model the tags —	146
9.5.3	How to model the communication interfaces —	148
9.5.4	Concept of LinkedObjects —	150
9.5.5	How to model the HMI —	150
9.5.6	How to model module services —	154
9.5.7	How to model alarms and events —	157
9.6	Industrial case studies, pilots and initiatives —	159
9.6.1	ACHEMA demonstrator —	159
9.6.2	Pilot for a biopharmaceutical project —	160
9.7	Discussion and Conclusion —	162
9.8	Outlook —	162
9.9	About the Authors —	164

Idar Pe Ingebrigtsen, Rainer Drath

10 AML domain model for System Control Diagrams — 165

10.1	Introduction —	165
10.1.1	The SCD origin —	165
10.1.2	What is a system control diagram (SCD)? —	166
10.1.3	Brief overview of the System Control Diagram (SCD) —	167
10.1.4	A detailed example – Level control —	168
10.1.5	Multidiscipline SCD development —	169
10.1.6	Input to the ICSS configuration —	170
10.2	The Scope of the AutomationML transfer —	170
10.2.1	The process part of system control diagrams (SCDs) —	170
10.2.2	The control part of system control diagrams (SCDs) —	171
10.3	The System Control Diagram Data model —	175
10.3.1	The data model —	175
10.3.2	Modelling strategy —	177
10.4	AML classes —	177
10.5	Industrial case studies, pilots and initiatives —	189
10.6	Discussion and Conclusion —	191
10.6.1	Document versus data-oriented design —	191
10.6.2	HMI display —	192
10.6.3	Future expansion of the SCD AML definition —	192
10.6.4	Cause and Effect documents —	192
10.6.5	Usage of vendor function blocks —	192
10.6.6	Strategy for handling unknown data —	193
10.7	Summary and outlook —	194
10.8	About the Authors —	195

Michael John

11 Data exchange between ECAD and PLC tools – AR APC — 197

11.1	Introduction and Motivation —	197
11.2	Typical Engineering Workflow —	198
11.3	APC Modelling Concept —	203

11.3.1	Neutral Data Model	203
11.3.2	Basic Requirements for the Implementation of the Neutral Model	204
11.3.3	Contents of Data Exchange	205
11.4	APC Data Exchange Information Model	206
11.5	AML Classes	209
11.5.1	Role class library AutomationProjectConfigurationRoleClassLib	209
11.5.2	Library AutomationProjectConfigurationInterfaceClassLib	210
11.6	APC Modelling Guideline	210
11.6.1	How to model an Automation Project	211
11.6.2	How to model a DeviceUserFolder	211
11.6.3	How to model a Subnet	211
11.6.4	How to model a Device	211
11.6.5	How to model a DeviceItem	212
11.6.6	How to model a TagTable	214
11.6.7	How to model a TagUserFolder	214
11.6.8	How to model a Tag	214
11.6.9	How to model a Channel	215
11.6.10	How to model a CommunicationInterface	215
11.6.11	How to model a Node	215
11.6.12	How to model a CommunicationPort	216
11.6.13	How to model an IOSystem	216
11.6.14	How to model a PowerPort	216
11.6.15	How to model a SensorPort	217
11.6.16	How to model a ModuleAssignment	217
11.6.17	How to model Bus specific Sections	217
11.6.18	How to model a CommunicationPortInterface	218
11.6.19	How to model a PowerPortInterface	218
11.6.20	How to model a SensorPortInterface	218
11.7	Example Application	218
11.7.1	Export from ECAD to AutomationML	219
11.7.2	Import from AutomationML into PLC	219
11.8	Discussion and Conclusion	220
11.9	About the Author	221

Matthias Mueller

12 Modelling of Drive Configurations – AR DRIVES MCAD — 223

12.1	Introduction and Motivation	223
12.1.1	Use Cases	223
12.1.2	Relation of electrical and mechanical aspects	225
12.2	AutomationML model for electrical aspects (ECAD)	226
12.2.1	Data model	226
12.2.2	AutomationML libraries	227
12.2.3	Modelling example	228
12.3	AutomationML model for mechanical aspects (MCAD)	229
12.3.1	Data model	229
12.3.2	AutomationML libraries	232
12.3.3	Modelling example	234
12.4	Discussion and Conclusion	235
12.5	About the Author	236

Ender Yemenicioglu

13 AML domain model for material handling — 237

- 13.1 Introduction and motivation — 237
- 13.2 Material Handling Toolchain — 238
- 13.3 AutomationML domain model for Material Handling — 240
 - 13.3.1 Module level (Level 1) — 240
 - 13.3.2 Port level (Level 2) — 243
- 13.4 Material Handling Libraries in AutomationML — 245
 - 13.4.1 Material Handling Interface Class Library — 245
 - 13.4.2 Role Class Library for Material Handling Resources — 245
 - 13.4.3 Role Class Library for Material Handling Processes — 247
 - 13.4.4 Role Class Library for Material Handling Products — 248
- 13.5 Application examples — 248
 - 13.5.1 Module level example — 248
 - 13.5.2 Port level example — 251
- 13.6 Conclusion — 253
- 13.7 About the Author — 253

Lorenz Hundt, Mathias Wiegand, Torben Meyer

14 The AutomationML Component — 255

- 14.1 Introduction and use cases — 255
 - 14.1.1 Automation Component — 255
 - 14.1.2 Stakeholder — 256
 - 14.1.3 Use Cases for AutomationML Components — 257
- 14.2 AutomationML Component data model — 265
 - 14.2.1 Structuring AutomationML Components in CAEX — 265
 - 14.2.2 Mapping of Aspects to AutomationML Elements — 267
 - 14.2.3 Mapping of general aspects of AutomationML Components — 269
 - 14.2.4 Role and Interface Class Libraries — 269
 - 14.2.5 Usage of role classes for model integration — 272
 - 14.2.6 Connecting AutomationML Components and Composite Components — 273
- 14.3 AMLX container for AutomationML Components — 276
- 14.4 Application example pneumatic cylinder ADN-25-50-A-P-A as AML Component — 277
 - 14.4.1 Step 0: Preconditions — 278
 - 14.4.2 Step 1: Modelling of an AML Component as SystemUnitClass — 279
 - 14.4.3 Step 2: How to model the attributes — 279
 - 14.4.4 Step 3: How to integrate graphic representation — 280
 - 14.4.5 Step 4: How to integrate documentations — 282
 - 14.4.6 Step 5: How to integrate behaviour — 283
 - 14.4.7 Step 6: How to integrate kinematics and geometry — 285
 - 14.4.8 Step 7: How to model connectors — 287
 - 14.4.9 Step 8: How to connect different model aspects — 289
 - 14.4.10 Step 9: How to store the AML Component as container package — 290
 - 14.4.11 Summary of the example — 290
- 14.5 Usability of AML Components for simulation in multidisciplinary tools chains — 292
- 14.6 Conclusion and Outlook — 293
- 14.7 About the Authors — 294

Rainer Drath, Markus Rentschler, Jörg Hinze

15 AML domain model for Electric Interfaces — 295

- 15.1 Introduction — 295
 - 15.1.1 Motivation — 295
 - 15.1.2 Use cases and value creation potential — 296
 - 15.1.3 Requirements — 297
- 15.2 Terms and modelling principles — 298
 - 15.2.1 Relation to other libraries — 299
 - 15.2.2 Overview of provided libraries — 300
- 15.3 AML attribute type libraries — 301
 - 15.3.1 Modelling of attribute semantics — 301
 - 15.3.2 GenericIndustrialAttributeTypeLib — 301
 - 15.3.3 GenericPinAttributeTypeLib — 302
 - 15.3.4 IEC61076-2 AttributeTypeLib — 302
 - 15.3.5 CableAttributeTypeLib — 303
- 15.4 Role class library for logic connector functions — 303
- 15.5 Interface class library for IEC61076-2 (M12, M8 and M5) — 304
 - 15.5.1 Overview — 304
 - 15.5.2 Attributes of the M12, M8 and M5 base classes — 305
 - 15.5.3 Attributes of the PinType base class — 306
- 15.6 Role class library for modelling cables — 306
- 15.7 Interface class library for IEC61076-2-101 — 307
 - 15.7.1 Library architecture and modelling concept — 307
 - 15.7.2 Connector variants for M12 and M8 connectors — 309
- 15.8 ANSI B93.55M library for 7/8" connectors — 310
- 15.9 IEC60603-7 library for RJ45 connectors — 310
- 15.10 IEC60807-2 library for Sub-D 9 connectors — 311
- 15.11 Library for USB connectors — 312
- 15.12 Library for terminal connectors — 312
- 15.13 General Modelling Rules — 313
- 15.14 Application examples — 315
 - 15.14.1 M12 to M12 cable — 315
 - 15.14.2 Automation component with multiple electric connectors — 316
 - 15.14.3 Automation topology — 318
- 15.15 Availability — 319
- 15.16 Conclusion and outlook — 320
- 15.17 About the Authors — 321

Markus Rentschler

16 AutomationML Component Checker — 323

- 16.1 Introduction and motivation — 323
- 16.2 Target groups and use cases — 324
- 16.3 Functionality — 324
- 16.4 Operating Environments — 325
- 16.5 AutomationML Component Checker user guide — 326
 - 16.5.1 Web based service interface — 326
 - 16.5.2 Login — 326
 - 16.5.3 File selection — 328
 - 16.5.4 Check — 329
 - 16.5.5 Test Report — 330

- 16.6 Summary — 333
- 16.7 About the Author — 334

Matthias Riedl, Holger Zipper

17 AML domain model for communication systems — 335

- 17.1 Introduction and use cases — 335
- 17.2 Object model of communication systems — 336
 - 17.2.1 General structure of a communication system — 336
 - 17.2.2 Model of a physical topology — 337
 - 17.2.3 Model of a logical topology — 338
 - 17.2.4 Model of a physical device — 338
- 17.3 Examples of communication systems — 339
- 17.4 AutomationML classes — 342
 - 17.4.1 The technology independent CommunicationRoleClassLib — 342
 - 17.4.2 The technology independent CommunicationInterfaceClassLib — 342
- 17.5 Modelling guideline for technology specific libraries — 343
 - 17.5.1 Application example for a physical network — 343
 - 17.5.2 How to model Profinet specific communication roles — 344
 - 17.5.3 How to model ProfNet specific communication interfaces — 344
 - 17.5.4 How to model a Profinet communication device type — 345
 - 17.5.5 How to integrate multiple communication devices to a physical communication system — 346
 - 17.5.6 How to model the logical communication system — 348
- 17.6 Discussion and Conclusion — 348
- 17.7 About the Authors — 349

Miriam Schleipen

18 Modelling OPC UA with AutomationML — 351

- 18.1 Why to combine AutomationML and OPC UA — 351
- 18.2 Who is involved? — 352
- 18.3 How to model an OPC UA server configuration in AML — 352
 - 18.3.1 Example — 352
 - 18.3.2 Modelling the OPC UA server — 353
 - 18.3.3 Step 2: Model necessary address space content of the OPC UA server — 358
 - 18.3.4 Step 3: Usage of the OPC UA server configuration information in an OPC UA client — 358
 - 18.3.5 Discussion: weakness of the BPR DataVariable — 359
- 18.4 How to use existing AML models in an OPC UA server (aka application of the companion specification AutomationML for OPC UA) — 359
- 18.5 Discussion and conclusion — 362
- 18.6 About the Author — 363

Arndt Lüder, Andreas Graf Gatterburg, Matthias Müller, Miriam Schleipen, Mathias Wiegand, Stefan Biffel, Rainer Drath

19 Serialization of the Asset Administration Shell by AutomationML — 365

- 19.1 Introduction and Use Cases — 365
 - 19.1.1 Rationales — 365
 - 19.1.2 Stakeholder and their Use Cases — 367
 - 19.1.3 Requirements for the AutomationML Model — 368

- 19.2 Meta model of Industry 4.0 Asset Administration Shell — 368
- 19.3 AAS modelling guideline with AutomationML — 370
 - 19.3.1 AutomationML classes for the AAS serialization — 370
 - 19.3.2 Guideline for structuring the AAS with AutomationML — 371
 - 19.3.3 Guideline for engineering tools — 371
- 19.4 Example for an AML based serialization of an Asset Administration Shell — 371
- 19.5 Information to be covered by Asset Administration Shells — 373
 - 19.5.1 Overview — 373
 - 19.5.2 Modelling guideline for submodels — 374
- 19.6 Exporting AutomationML by the AASX Package Editor — 375
- 19.7 When to use AML in the context of the AAS — 375
- 19.8 Summary — 376
- 19.9 About the Authors — 377

Martin Langosch, Roman Mader

20 AutomationML Industrialization and Toolchain — 379

- 20.1 Introduction and Motivation — 379
- 20.2 Approach — 381
- 20.3 Data exchange between Tools and Platforms — 382
- 20.4 Data governance — 384
- 20.5 About the Authors — 384

Nicole Schmidt, Matthias Rassl

21 AutomationML governance at Daimler AG — 385

- 21.1 Introduction — 385
- 21.2 What is an AutomationML governance — 386
- 21.3 Application of the AutomationML governance at Daimler AG — 388
 - 21.3.1 What is addressed by the AutomationML governance? — 388
 - 21.3.2 Who is involved in the AutomationML governance? — 390
 - 21.3.3 How does the AutomationML governance work? — 392
- 21.4 Discussion and Conclusion — 394
- 21.5 About the Authors — 396

Michael John

22 AutomationML and eCl@ss Integration — 397

- 22.1 Motivation and Scope — 397
- 22.2 Use Cases and Considered Data Exchange Structures — 398
 - 22.2.1 Technical Use Cases — 398
 - 22.2.2 Exchange Use Cases — 400
 - 22.2.3 Application Use Cases — 403
- 22.3 Realization of Technical Use Cases — 405
 - 22.3.1 How to integrate eCl@ss Attribute Semantics via IRDI — 405
 - 22.3.2 How to integrate eCl@ss Object Semantics via eClassClassSpecification RoleClass — 406
 - 22.3.3 How to generate an eCl@ss Role Class Library — 407
 - 22.3.4 How to transform eCl@ss Attributes — 409
- 22.4 Example — 409
- 22.5 Discussion and Conclusion — 410
- 22.6 About the Author — 410

Prerna Juhlin, Rainer Drath

23 Semantic and Pragmatic Interoperability Mappings — 411

- 23.1 Introduction — 411
- 23.2 How to exchange data without semantic standardization — 412
- 23.3 Overview of the methodology — 414
 - 23.3.1 Phase 1: Interoperability Mappings Creation — 414
 - 23.3.2 Phase 2: Automatic Data Exchange with Meaning and Usage Alignments — 418
- 23.4 Modelling guideline — 421
 - 23.4.1 Application Example — 421
 - 23.4.2 Technical approach — 422
 - 23.4.3 Modelling of Data Object Types as SystemUnitClass (Phase 1, Step 1) — 425
 - 23.4.4 How to model Semantic Interoperability Alignments (Phase 1, Step 2) — 425
 - 23.4.5 From Semantic Mappings to Semantic Standardization (Phase 1, Step 2') — 429
 - 23.4.6 How to model Pragmatic Interoperability Alignments (Phase 1, Step 3) — 430
 - 23.4.7 Automatic Data Exchange with Meaning and Usage Alignments (Phase 2) — 432
- 23.5 Discussion and Conclusion — 433
- 23.6 About the Authors — 434

Miriam Schleipen

24 Extended RoleClass libraries — 435

- 24.1 Introduction: the value of RoleClass libraries — 435
- 24.2 How to use AutomationML RoleClass Libraries — 436
 - 24.2.1 User-defined RoleClass libraries in one model — 436
 - 24.2.2 User-defined RoleClass libraries in multiple models — 437
 - 24.2.3 User-defined RoleClass libraries in a defined context — 437
 - 24.2.4 AutomationMLBaseRoleClassLib — 437
 - 24.2.5 Normative AutomationML IEC RoleClass libraries — 438
 - 24.2.6 Informative AutomationML IEC RoleClass libraries — 439
 - 24.2.7 AutomationML e.V. RoleClass libraries — 440
- 24.3 How to find and import AutomationML RoleClass Libraries — 441
- 24.4 How to model a plant hierarchy with the ResourceStructure semantic — 443
 - 24.4.1 Example — 445
 - 24.4.2 Step 1: Create an InstanceHierarchy with InternalElements — 446
 - 24.4.3 Step 2: Apply the semantics (aka RoleClasses) — 447
- 24.5 The ISA 106 hierarchy as example how to map a user-defined RoleClass library — 447
- 24.6 Discussion and conclusion — 448
- 24.7 About the author — 449

Bernhard Wally, Arndt Lüder

25 AML-based Enterprise Control System Integration by IEC 62264 — 451

- 25.1 Introduction and Use Cases — 451
- 25.2 Data Model — 453
 - 25.2.1 Personnel — 454
 - 25.2.2 Equipment — 455
 - 25.2.3 Material — 455
 - 25.2.4 Process Segments — 456
 - 25.2.5 Further Concepts — 456
- 25.3 AML Classes — 456
- 25.4 Modelling Guideline — 458

- 25.4.1 Application Example — 458
- 25.4.2 How to Model Personnel — 459
- 25.4.3 How to Model Equipment — 460
- 25.4.4 How to model Material — 461
- 25.4.5 How to Model Process Segments — 461
- 25.5 Discussion and Conclusion — 464
- 25.6 About the Authors — 465

References for Part II — 467

Part III: The Future – AutomationML Research — 481

Rainer Drath

26 Overview of Part III — 483

Birgit Vogel-Heuser, Minjie Zou, Michael Sollfrank

27 AutomationML based development of mechatronic systems — 485

- 27.1 Introduction and Motivation — 485
- 27.2 Problems in the Workflow of Automated Production Systems — 486
- 27.3 The research approach: Model-Based Systems Engineering — 486
 - 27.3.1 About Model-Based Systems Engineering (MBSE) — 486
 - 27.3.2 Coupling the industrial toolchain in mechatronic MBSE via AutomationML: The Architecture — 487
 - 27.3.3 Mapping between SysML4Mechatronics and AutomationML — 489
 - 27.3.4 Generation of MCAD from AutomationML — 491
 - 27.3.5 Generation of ECAD from AutomationML — 493
 - 27.3.6 Generation of the PLC Control Program from AutomationML — 495
 - 27.3.7 Inconsistency Identification based on AutomationML — 496
- 27.4 The overall MBSE Workflow Enabled by AutomationML — 497
- 27.5 Discussion and Conclusion — 499
- 27.6 References — 501
- 27.7 About the Authors — 502

Birgit Vogel-Heuser, Felix Ocker, Lukas Gehlen

28 Concept to refine and computationally evaluate PPR information in AML — 503

- 28.1 Introduction and Use Case — 503
 - 28.1.1 Application Example — 504
 - 28.1.2 Modelling Products, Processes, and Resources of the xPPU — 505
 - 28.1.3 Research questions and requirements — 507
- 28.2 Approach — 507
 - 28.2.1 Refined interface definitions — 507
 - 28.2.2 Representation of Changes to the Plant in AML — 512
 - 28.2.3 Extracting Information from AML — 513
- 28.3 Discussion and Conclusion — 517
- 28.4 References — 517
- 28.5 About the Authors — 519

Dominik Braun, Behrang Ashtari, Michael Weyrich

29 Integration of data and software into the Digital Twin via AML — 521

- 29.1 Introduction — 521
- 29.2 Characteristics of the Digital Twin — 522
- 29.3 Tools and Models integration approaches for Digital Twins — 524
- 29.4 An Example for the Creation of a Digital Twin using PLM systems — 525
- 29.5 Reconfiguration of automated Systems with a Digital Twin — 529
- 29.6 Challenges of structured models within PLM systems — 530
- 29.7 AutomationML increasing the flexibility of Digital Twins — 531
- 29.8 Towards Future research about the combination of PLM and AML for Digital Twin — 534
- 29.9 Conclusion and Outlook — 535
- 29.10 References — 536
- 29.11 About the Authors — 538

Andreas Würger

30 Optimizing the engineering of technical energy management systems — 539

- 30.1 Introduction — 539
- 30.2 An engineering concept for technical energy management systems — 540
- 30.3 Relevant scientific papers and publications — 542
 - 30.3.1 Hardware configuration modelling — 542
 - 30.3.2 Modelling plants from an energy point of view — 543
 - 30.3.3 A combination of AutomationML and OPC UA — 543
- 30.4 Modelling of PROFINET communication systems — 544
- 30.5 Concept for Energy Information Models — 545
- 30.6 Concept for an Energetic Production Site Model — 547
 - 30.6.1 Definition and modelling of physical balancing groups — 548
 - 30.6.2 Definition and modelling of user-defined balancing groups — 549
 - 30.6.3 Energetic Production Site Model in AML — 550
- 30.7 Evaluation and outlook — 551
- 30.8 References — 551
- 30.9 About the Author — 553

Milan Vathoopan, Kirill Dorofeev, Alois Zoitl

31 Skill-Based Engineering of Automation Systems: Use Case and Evaluation — 555

- 31.1 Introduction — 555
- 31.2 Background — 556
 - 31.2.1 Benefit of Skill-Based Engineering on Current Automation Systems — 556
 - 31.2.2 State of the Art in Skill-based Automation Engineering and Operation — 557
- 31.3 Model of a Skill — 558
- 31.4 Realizing a Skill-based Automation System — 561
- 31.5 Automation System Planning with SystemPlanner — 563
 - 31.5.1 Basic Methodology of Planning Used in SystemPlanner — 563
 - 31.5.2 Prototypical Implementation — 565
- 31.6 Generating Executable Skill-based Control Logic — 567
- 31.7 Evaluation — 569
 - 31.7.1 Skill-Based Planning of the Pick-and-Place Station — 570
 - 31.7.2 Generating the code — 572

31.8	Outlook and Conclusion —	573
31.9	References —	574
31.10	About the Authors —	577

Arndt Lüder, Stefan Biffl, Felix Rinker, Anna-Kristin Behnert

32 Engineering Data Logistics based on AML — 579

32.1	Introduction —	579
32.2	Stakeholder and their Use Cases —	581
32.3	Running example —	582
32.4	Basic data logistics architecture —	583
32.5	Prerequisites for application – common concept identification and modelling —	586
32.5.1	Common Concepts Identification —	586
32.5.2	AutomationML representation of concepts —	588
32.6	Flexible tool adapters —	592
32.6.1	CSV adapter —	592
32.6.2	XML adapter —	593
32.7	Central data logistics —	594
32.8	Application example —	595
32.9	Software system requirements —	598
32.10	Summary —	599
32.11	References —	600
32.12	About the Authors —	601

Janni Sinnemann, Matthias Bartelt, Anton Strahilov, Dimitri Penner, Bernd Kuhlenkötter

33 Energy optimization during virtual commissioning — 603

33.1	Virtual Commissioning Workflow —	603
33.1.1	Introduction —	603
33.1.2	Problems and objectives —	606
33.1.3	Fields of action and use of AutomationML —	606
33.2	Modelling recommendations —	608
33.2.1	How to model digital twins —	608
33.2.2	How to manage the versioning of engineering states —	612
33.2.3	How to model production processes —	615
33.2.4	How to model energy profiles —	621
33.2.5	How to connect different planning data —	623
33.3	Discussion and Conclusion —	627
33.4	References —	628
33.5	About the Authors —	629

Abbreviations — 631

Trademarks — 635

Index — 637