

# Contents

Foreword by Peter Lieber	xii
Foreword to the English edition	xiii
Foreword to the German edition	xv
<b>1 Introduction</b>	<b>1</b>
1.1 Who should read this book? . . . . .	3
1.2 What's next? . . . . .	4
1.3 Website for this book . . . . .	4
<b>I Fundamentals</b>	<b>5</b>
<b>2 Systems Engineering</b>	<b>7</b>
2.1 What is a System? . . . . .	7
2.2 Systems Engineering . . . . .	8
2.2.1 Architecture . . . . .	9
2.2.2 Requirements . . . . .	10
2.2.3 System behavior . . . . .	14
2.3 The Systems Engineering Schema . . . . .	15
<b>3 Model-Based Development</b>	<b>19</b>
3.1 Model, abstraction and view . . . . .	19
3.2 Model-Driven Architecture . . . . .	21
3.3 Metamodels . . . . .	23

3.4	Domain-specific modeling . . . . .	25
3.5	Profiles . . . . .	25
3.6	Work products of model-based development . . . . .	27
<b>4</b>	<b>SysML</b>	<b>29</b>
4.1	What is SysML? . . . . .	29
4.2	SysML is the basis of systems development . . . . .	30
4.3	SysML and UML . . . . .	31
4.4	Basic concepts of object orientation . . . . .	32
4.4.1	Classes and Objects . . . . .	32
4.4.2	Inheritance . . . . .	34
4.4.3	Special Instances: Parts . . . . .	36
4.4.4	Blocks and Properties . . . . .	36
4.5	Separation of model and view . . . . .	39
4.6	SysML diagrams . . . . .	40
4.6.1	Diagram frames . . . . .	41
4.6.2	The Package diagram . . . . .	42
4.6.3	The Block Definition diagram . . . . .	43
4.6.4	The Internal Block diagram . . . . .	46
4.6.5	The Parametric diagram . . . . .	47
4.6.6	The Use Case diagram . . . . .	49
4.6.7	The Requirements diagram . . . . .	51
4.6.8	The Sequence diagram . . . . .	53
4.6.9	The Activity diagram . . . . .	54
4.6.10	The State diagram . . . . .	58
4.7	More SysML Constructs . . . . .	59
4.7.1	The Allocation . . . . .	59
4.7.2	Viewpoints and Views . . . . .	60
4.7.3	Profiles . . . . .	62
4.7.4	Elements that appear only in diagrams and not in models . . . . .	62
4.8	What SysML isn't... . . . .	63
<b>II</b>	<b>Practical Use</b>	<b>65</b>
<b>5</b>	<b>Tooling</b>	<b>67</b>
5.1	Criteria for tool selection . . . . .	68

---

5.2	Tool infrastructure . . . . .	70
5.3	Tool testing and release . . . . .	71
5.4	Enterprise Architect . . . . .	71
5.4.1	Editing the models . . . . .	74
5.4.2	Advanced features . . . . .	74
5.4.3	The role of add-ins and tools in the context of development . . . . .	82
<b>6</b>	<b>Definition of Development Context</b>	<b>83</b>
6.1	Processes are absolutely necessary . . . . .	83
6.2	The standard V-model . . . . .	85
6.3	Process models and development standards . . . . .	86
6.3.1	CMMI, SPICE and Automotive SPICE . . . . .	87
6.3.2	The INCOSE Systems Engineering Handbook . . . . .	89
6.3.3	IEC 61508 und ISO 26262 . . . . .	90
6.4	Functional and technical development . . . . .	91
6.4.1	Functional development . . . . .	92
6.4.2	Technical development . . . . .	94
6.5	Architectural construction kit . . . . .	101
6.6	Abstraction Levels . . . . .	102
6.7	Validation and verification . . . . .	105
6.8	Traceability . . . . .	106
<b>7</b>	<b>Implementation Example</b>	<b>109</b>
7.1	A new development request . . . . .	109
7.2	A first system context definition . . . . .	110
7.3	Technical control chain model . . . . .	113
7.3.1	Encapsulation of components . . . . .	114
7.3.2	Decomposition view . . . . .	117
7.3.3	Architecture-based requirements determination . . . . .	118
7.3.4	Integrating the test . . . . .	120
7.4	Physical model . . . . .	122
7.5	Allocation . . . . .	124
7.6	Extended customer requirements . . . . .	125
7.6.1	Technical control chain model . . . . .	126
7.6.2	Physical model . . . . .	127
7.7	Behavior modeling . . . . .	127
7.8	Conclusion . . . . .	132

<b>8</b>	<b>Supporting Processes and Concepts</b>	<b>135</b>
8.1	Versioning and baselining . . . . .	135
8.1.1	Versioning and baselining of models . . . . .	136
8.1.2	Versioning of auxilliary tools . . . . .	138
8.2	Concepts of re-use . . . . .	139
8.3	Variant management . . . . .	141
8.3.1	Feature modeling . . . . .	142
8.3.2	Variant generation . . . . .	143
8.4	Tool integration . . . . .	146
8.4.1	Integration or requirements . . . . .	146
8.4.2	Involvement of FMEA . . . . .	147
8.4.3	Integration of function-oriented development . . . . .	150
8.4.4	Open Services for Lifecycle Collaboration . . . . .	151
8.5	Document generation . . . . .	152
8.6	Model validation and metrics . . . . .	154
8.6.1	Formal modeling rules . . . . .	154
8.6.2	Metrics . . . . .	154
8.7	Automotive Open Systems Architecture . . . . .	156
8.8	Industry 4.0 . . . . .	158
8.8.1	The vision behind Industry 4.0 . . . . .	158
8.9	Code generation and UML-level debugging . . . . .	159
8.9.1	Model-level debugging . . . . .	160
8.10	Model data access with customized user interfaces . . . . .	160
<b>9</b>	<b>Model Details</b>	<b>161</b>
9.1	Model structure . . . . .	161
9.2	Separating the system into modules . . . . .	165
<b>10</b>	<b>Deployment of Model-Based Systems Engineering</b>	<b>169</b>
10.1	Paradigm shift needed . . . . .	169
10.2	Management support . . . . .	170
10.3	Occupying roles with the right employees . . . . .	171
10.4	Training . . . . .	171
10.5	Seamless tool chain . . . . .	172
10.6	Practical experience is important . . . . .	173
<b>11</b>	<b>Outlook</b>	<b>175</b>

---

11.1	Metamodeling . . . . .	175
11.2	Model transformation . . . . .	179
11.2.1	QVT . . . . .	180
11.2.2	Model to Text transformation . . . . .	184
11.3	Object Constraint Language . . . . .	186
11.4	Model simulation . . . . .	187
11.5	Model-based testing . . . . .	187
11.6	Model visualization as city map . . . . .	187
11.7	Strong links between requirements and architecture . . . . .	189
11.8	Big data . . . . .	189
11.9	Use of new user interfaces . . . . .	190
11.10	Conclusion . . . . .	191
<b>A</b>	<b>Modeling Rules</b>	<b>193</b>
<b>B</b>	<b>Classification in SPICE</b>	<b>197</b>
<b>C</b>	<b>About LieberLieber</b>	<b>201</b>
<b>D</b>	<b>System Modeling Quick Reference</b>	<b>205</b>
	<b>References</b>	<b>211</b>
	<b>List of Figures</b>	<b>216</b>
	<b>Index</b>	<b>221</b>