



# Contents

<b>Preface</b>	<b>xv</b>
<b>Acknowledgments</b>	<b>xxiii</b>
<b>Foreword</b>	<b>xxvii</b>
<b>Part I Introduction to Software Factories</b>	<b>1</b>
<b>Chapter 1 Introduction</b>	<b>3</b>
Tools Lag Platforms	4
The Software Development Landscape	7
Platform Technology Evolution	8
Building Applications with Services	24
Software Development Challenges	30
Discontinuous Innovation	32
Notes	34
<b>Chapter 2 Dealing with Complexity</b>	<b>35</b>
The Problem of Complexity	35
Feature Delocalization	38
Working at the Wrong Level of Abstraction	41
Raising the Level of Abstraction	53
Packaging Abstractions	61
Current Methods and Practices	66
Encapsulation	67
Partitioning Responsibility	68
Documenting Design	73
Automating Development	77
Organizing Development	82
Notes	85

## viii Contents

---

<b>Chapter 3</b>	<b>Dealing with Change</b>	<b>87</b>
	The Problem of Change	87
	Responses to Change	87
	Software Aging	88
	Software As Simulation	90
	Sources of Change	99
	Current Methods and Practices	100
	Preventing Stagnation	101
	Reducing Brittleness	104
	Reducing Fatigue	106
<b>Chapter 4</b>	<b>Paradigm Shift</b>	<b>109</b>
	Chronic Problems	109
	Monolithic Construction	110
	Gratuitous Generality	115
	One-Off Development	120
	Process Immaturity	121
	Critical Innovations	125
	Systematic Reuse	125
	Development by Assembly	129
	Model-Driven Development	139
	Process Frameworks	150
<b>Chapter 5</b>	<b>Software Factories</b>	<b>155</b>
	Industrializing Software Development	155
	The Economics of Reuse	156
	Economies of Scale and Scope	157
	Systematic Software Reuse	159
	Integrating the Critical Innovations	161
	What Is a Software Factory?	163
	What Is a Software Factory Schema?	164
	What Is a Software Factory Template?	173
	Building a Software Factory	174
	Building a Software Product	175
	A Software Factory Example	180
	Implications of Software Factories	185
	Development by Assembly	185
	Software Supply Chains	186
	Relationship Management	187
	Domain Specific Assets	187
	Organizational Changes	188
	Mass Customization of Software	188
	Realizing the Software Factory Vision	189
	Notes	190

---

<b>Part II</b>	<b>Critical Innovations</b>	<b>191</b>
<b>Chapter 6</b>	<b>Models and Patterns</b>	<b>193</b>
	Intent versus Implementation	193
	What Is Intent?	194
	Capturing Intent	195
	A Closer Look at Patterns	197
	Creating and Using Patterns	198
	Weaving Patterns into Languages	201
	Patterns in Product Families	204
	Encapsulating Pattern Languages	205
	What Is Encapsulation?	206
	Defining Languages with Encapsulation	208
	Formalizing Pattern Languages	210
	What Are Models?	212
	Models as Abstractions	213
	Model Visualization	214
	Models as Metadata	217
	Models as Development Artifacts	218
	Modeling or Programming?	226
<b>Chapter 7</b>	<b>Programming with Models</b>	<b>231</b>
	Model-Driven Development	231
	Generating Software	233
	Automating Patterns	238
	Automating Refactoring	249
	Automating Builds	249
	Automating Deployment	249
	Automating Testing	251
	Debugging with Models	252
	Using Multiple Views	253
	Architectural Description	253
	Domain Specificity	254
	Modeling Aspects	258
	How to Model Software	263
	Types of Information	264
	Level of Abstraction	266
	Style of Specification	271
	Domain-Specific Languages	274
	Business Ramifications	274
	Technical Ramifications	275
	Next Steps	277
	Notes	277
<b>Chapter 8</b>	<b>Language Anatomy</b>	<b>279</b>
	Example Language	282

**x Contents**

---

Abstract Syntax	285
Context-Free Grammars and BNF	286
Metamodels	289
Abstract Syntax Graphs	293
Well-Formedness Rules	294
Comparison of CFGs and Metamodels	297
Concrete Syntax	299
Serialization Syntax	306
Semantics	309
Translational Semantics	310
Trace-Based Semantics	312
Programming versus Modeling	318
Notes	319
<b>Chapter 9 Families of Languages</b>	<b>321</b>
Language Families	321
Tool Factories	328
Tool Factory Architecture	328
The State of the Art	333
<b>Chapter 10 Systematic Reuse</b>	<b>337</b>
Software Product Families	337
Software as a Product	338
How Families Are Formed	338
Working Within a Family	340
Solving Problems in Advance	341
The Role of Architecture	342
Commonality and Variability	344
Where Families Are Found	345
Software Product Lines	347
Creating Economies of Scope	348
Required Adaptations	350
Software Supply Chains	355
Notes	358
<b>Chapter 11 Software Product Lines</b>	<b>359</b>
Product Line Development	359
Product Line Analysis	361
Product Line Design	375
Product Line Implementation	379
Product Development	382
Problem Analysis	382
Product Specification	382
Collateral Development	385
Product Implementation	385
Product Line Evolution	391
Product Line Deployment	391

---

<b>Chapter 12</b>	<b>Platform-Based Abstractions</b>	<b>393</b>
	Platform-Based Abstractions	393
	Platform-Based Abstractions in Product Lines	395
	Properties of Platform-Based Abstractions	397
	Classes, Libraries, and Frameworks	401
	Class Frameworks	403
	Summary	411
<b>Chapter 13</b>	<b>Components and Services</b>	<b>413</b>
	Software Components	413
	Definition	413
	Interface Specifications	415
	Component Dependencies	423
	Component-Based Development	426
	Defining Product Line Component Architectures	432
	From Components to Services	438
	Service Components	438
	Business Protocols and Contracts	439
	Service-Oriented Architectures	445
	Web Services Technology	446
	What's Different This Time?	448
	Web Service Data Semantics	448
	WS Process Semantics	450
	Summary	452
	Notes	453
<b>Chapter 14</b>	<b>Mappings and Transformations</b>	<b>455</b>
	Transformations	455
	Types of Transformation	456
	Vertical Transformations	458
	Horizontal Transformations	462
	Transformation Problems	470
	Model-to-Model Transformations	471
	Model-to-Code Transformations	473
	Solving the Composition Problem	476
	Solving the Traceability Problem	479
	Summary	481
	Notes	481
<b>Chapter 15</b>	<b>Generating Implementations</b>	<b>483</b>
	Describing Transformations	483
	Mapping Rules Have IF-THEN Parts	485
	Implementing Mapping Rules	489
	Specifying Horizontal Transformations	493
	Describing Aspect Weaving	493
	Describing Refactoring Rules	495
	Patterns as Sets of Mapping Rules	496

**xii Contents**

---

	Transformation Systems	500
	Black-Box and White-Box Transformations	501
	Grey-Box Transformation Systems	503
	Applying Black-Box Transformations	505
	Summary	508
<b>Part III</b>	<b>Software Factories in Depth</b>	<b>509</b>
<b>Chapter 16</b>	<b>A Software Factory Example</b>	<b>511</b>
	A Review of the Approach	511
	Building an Online eCommerce	
	Application Family	512
	Product Line Analysis	515
	Product Line Definition	515
	Problem Domain Scoping	518
	Solution Domain Scoping	521
	Business Case Analysis	524
	Product Line Design	526
	Product Line Architecture Development	526
	Product Line Requirements Mapping	546
	Product Line Implementation	548
	Asset Provisioning	548
	Asset Packaging	557
	Product Development	557
	Product Specification	558
	Conclusion	560
	Notes	561
<b>Chapter 17</b>	<b>Frequently Asked Questions</b>	<b>563</b>
	How Do Software Factories Differ From . . . ?	563
	How Do Software Factories Differ from RAD?	564
	How Do Software Factories Differ	
	from MDA <sup>®</sup> ?	567
	How Do Software Factories	
	Differ from the UP?	573
	How Do Software Factories Differ	
	from Agile Modeling?	577
	How Agile Are Software Factories?	580
	Agile Development Principles	580
	Agile Development Practices	582
	How Will Software Factories Be Adopted?	584
	Types of Business Applications	586
	Business Application Characteristics	586
	How Mature Are Software Factories?	588
	Language Technology	589
	Tool Extensibility	589
	Pattern Composition	590

---

**Contents** **xiii**

Deferred Encapsulation	590
Standard Assets for Popular Domains	591
How Should Software Factories Be Implemented?	591
New Development Artifacts	592
New Development Scope	592
New Development Activities	592
Implementing Software Factories	593
What's Different This Time?	594
<b>Appendix A: Abstraction and Refinement</b>	<b>597</b>
<b>Appendix B: The Unified Modeling Language</b>	<b>611</b>
<b>Bibliography</b>	<b>629</b>
<b>Index</b>	<b>641</b>

