

Contents

1	Introducing Requirements-Driven Modelling	1
1.1	Why Model Requirements?	2
1.2	Making Requirements Precise	9
1.2.1	Writing Good Stories	9
1.2.2	Writing Stories About Software	10
1.2.3	How About Quality Issues?	15
1.3	What Is the Meaning of Requirements Models?	16
1.3.1	Requirements Explained Through Observable Behaviour	16
1.3.2	Requirements Explained Through Translation into Java	20
1.4	Towards Model-Driven Requirements Engineering	22
1.4.1	“Traditional” Software Development	22
1.4.2	Model-Driven Software Development	24
1.4.3	Software Development with DSLs and Model-Driven Requirements	27
2	Presenting the Requirements Specification Language	31
2.1	How to Define a Modelling Language?	31
2.2	Structuring Requirements Specifications	33
2.2.1	Basic Concepts	34
2.2.2	Packaging and Presenting Requirements	36
2.3	Specifying the Problem Domains and Their Rules	39
2.3.1	Defining the Problem Domain	40
2.3.2	Defining the Application Domain	43
2.3.3	Defining the Domain Rules	45
2.4	Specifying Functional Requirements	48
2.4.1	Use Cases and Relationships	49
2.4.2	Sentence Types	55
2.4.3	Scenarios	59

3	Defining RSL	67
3.1	Introduction to Metamodelling	67
3.2	Overview of the RSL Metamodel	72
3.3	Terms and Phrases	74
3.4	Domain Elements and Relationships	78
3.5	Constrained Language Sentences and Scenarios	82
3.6	Requirements and Use Cases	86
3.7	Domain and Requirements Specifications	91
3.8	Summary of Metamodelling	95
4	Explaining RSL with Java	101
4.1	Translational Framework	101
4.2	Semantics Involving the General Structure	113
4.3	Semantics Involving the View Layer	119
4.4	Semantics Involving the Presenter and Model Layers	127
4.5	Summary Example	140
5	Understanding Model Transformations	149
5.1	Overview	149
5.2	“Hello World” in MOLA—Declarative Processing	153
5.3	Variables and Procedures in MOLA—Imperative Processing	157
5.4	More Advanced MOLA Constructs	162
5.5	End-to-End Transformation Example	169
5.6	Which Language to Choose?	181
6	Writing Model Transformations for Requirements	185
6.1	Using the MOLA Tools	185
6.1.1	Specifying the Metamodel	186
6.1.2	Specifying and Compiling Transformation Programs	188
6.1.3	Debugging Transformation Programs	191
6.2	Transformation Overview	194
6.3	Generation of the Basic Structure	197
6.4	Generation of Data Transfer Objects	204
6.5	Parsing of Use Case Scenarios	210
6.6	Generation of the Presenter Layer Details	215
7	Applying MDRE in Practice	225
7.1	Using the ReDSeeDS Tool	225
7.2	Introducing the ReDSeeDS Methodology	232
7.2.1	Overview of the ReDSeeDS Process	233
7.2.2	Software System Evolution Process	236
7.2.3	Technology Evolution Process	241

7.3	Reuse Approaches with Requirements Models	243
7.3.1	Applying MDRE to Existing (Legacy) Systems	243
7.3.2	Reusing Requirements Models Through Patterns	249
7.4	Summary: Is MDRE for Me?	254
8	Case Study	257
8.1	Study Assumptions and Context	257
8.2	Source Model in RSL	259
8.2.1	General Structure	259
8.2.2	Use Case Representation Details	262
8.3	General Architecture of the Generated System	272
8.4	User Interface Code	280
8.5	Application Logic Code	288
8.6	Discussion	293
	Appendix A: Summary of RSL Syntax	297
	Appendix B: Summary of MOLA Syntax	317
	Literature	333
	Index	343

From Requirements to Java in a Snap
Model-Driven Requirements Engineering in Practice
Smialek, M.; Nowakowski, W.
2015, XXIII, 352 p. 295 illus., Hardcover
ISBN: 978-3-319-12837-5