

Contents

1	Introduction	1
1.1	Research Motivation	4
1.2	Problem Statement	6
1.3	Thesis Contributions	8
1.4	Research Method	12
1.5	Context of the Thesis	13
1.6	Outline	13
2	Background and Technological Context	17
2.1	Method Engineering	18
2.1.1	Defining Method Engineering	18
2.1.1.1	Terminology	21
2.1.2	Method Design	22
2.1.2.1	The SPEM 2.0 Standard	23
2.1.2.2	The BPMN 2.0 Standard	26
2.1.3	Method Implementation	29
2.1.3.1	Computer-Aided Software Engineering	30
2.1.3.2	MetaCASE Environments	31
2.1.4	Method Execution	33
2.1.4.1	Operational Aspects of BPMN 2.0	33
2.2	Model-Driven Engineering	34
2.2.1	Defining Model-Driven Engineering	35

2.2.2	Metamodeling	36
2.2.2.1	Domain-Specific Languages	37
2.2.3	Model Transformations	38
2.2.4	Models at Runtime	41
2.3	Eclipse-based Technologies	44
2.3.1	The Eclipse Platform	44
2.3.2	Eclipse Modeling	47
2.3.3	Eclipse Process Framework	49
2.3.4	Activiti	50
2.3.5	MOSKitt	51
2.4	Conclusions	53
3	State of the Art	55
3.1	Properties to Analyze Method Engineering Approaches	56
3.2	Method Engineering Approaches	58
3.2.1	Assembly-based	60
3.2.1.1	Brinkkemper <i>et al.</i> Approach	61
3.2.1.2	Prakash Approach	65
3.2.1.3	Ralyté <i>et al.</i> Approach	69
3.2.1.4	OPEN Process Framework	74
3.2.1.5	Method Editor	77
3.2.2	Paradigm-based	80
3.2.2.1	Rolland <i>et al.</i> Approach	80
3.2.2.2	MetaEdit+	84
3.2.2.3	Ralyté <i>et al.</i> Approach	88
3.2.2.4	Work Product Pool Approach	91
3.2.3	Configuration-based	94
3.2.3.1	Deneckère <i>et al.</i> Approach	95
3.2.3.2	Method for Method Configuration	99
3.2.3.3	Process Configuration Approach	106

3.3	Conclusions	109
4	A Model-Driven Approach for Method Engineering	115
4.1	Overview	117
4.1.1	Origins: The MOSKitt Project	117
4.1.2	Developing the Methodological Approach	118
4.1.3	Phases of the Approach	119
4.2	Method Design	121
4.2.1	A DSL for the Conceptual Modeling of Methods	122
4.2.1.1	The MOF Metamodel	123
4.2.1.2	Supporting the Method Product and Process Parts	125
4.2.2	Promoting Reuse: A Method Fragment Taxonomy	126
4.2.2.1	Fragment Types	128
4.2.2.2	Fragment Structure	131
4.2.3	A Process for Method Design	132
4.2.3.1	Method Definition	133
4.2.3.2	Method Configuration	135
4.2.3.3	Executable Process Generation	138
4.3	Method Implementation	141
4.3.1	An Automatic Process for CASE Environment Construction	143
4.3.1.1	Identification of Software Tools	144
4.3.1.2	Resolution of Dependencies	147
4.3.1.3	Deployment of Software Tools	148
4.4	Method Execution	150
4.4.1	The Project Manager Component	150
4.4.2	Method Process Support	153
4.4.3	Method Product Support	154
4.5	Conclusions	157
5	MOSKitt4ME: A Software Infrastructure	159
5.1	Functional Requirements	160

5.2	Developing MOSKitt4ME	162
5.2.1	The Architecture of MOSKitt4ME	163
5.2.2	Implementation of the Architecture	165
5.3	The MOSKitt4ME Environment	168
5.3.1	The CAME Part of MOSKitt4ME	169
5.3.1.1	Specifying Methods: the Method Editor	169
5.3.1.2	Connecting to Repositories: the Repository Client	172
5.3.1.3	Building Eclipse Plug-ins: the Metatools	186
5.3.1.4	Obtaining Software Support: the CASE Generator	189
5.3.2	The CASE Part of MOSKitt4ME	191
5.3.2.1	Bringing Methods into Enactment: the Project Manager	192
5.4	Conclusions	200
6	Evaluation of the Proposal	203
6.1	Background on Method Engineering Evaluation	204
6.2	Overview of the Evaluation Study	206
6.2.1	Measures of Usefulness and Ease of Use	206
6.2.2	Experimental Process	208
6.3	Definition and Planning	208
6.3.1	Goal	208
6.3.2	Research Questions	209
6.3.3	Subjects	209
6.3.4	Object	210
6.3.5	Factors and Treatments	211
6.3.6	Tasks	211
6.3.7	Context	213
6.3.8	Instrumentation	213
6.3.9	Experimental Setup	214
6.3.10	Validity Evaluation	215
6.3.10.1	Conclusion Validity	215

6.3.10.2	Internal Validity	216
6.3.10.3	Construct Validity	216
6.3.10.4	External Validity	217
6.4	Execution	217
6.4.1	Preparation	217
6.4.2	Operation	219
6.4.3	Data Validation	219
6.5	Data Analysis	220
6.5.1	Analysis of the Subjective Data	220
6.5.1.1	Quantitative Feedback	221
6.5.1.2	Qualitative Feedback	221
6.5.1.3	Mental Effort	221
6.5.2	Analysis of the Objective Data	222
6.5.2.1	Session Transcription	223
6.5.2.2	Coding Scheme Definition	223
6.5.2.3	Protocol Construction	225
6.5.2.4	Protocol Analysis	226
6.6	Results	227
6.6.1	Research Question 1	227
6.6.2	Research Question 2	229
6.6.3	Research Question 3	232
6.6.3.1	Efficiency	232
6.6.3.2	Effectiveness	234
6.6.4	Research Question 4	234
6.6.4.1	Difficulty Using SPEM 2.0	235
6.6.4.2	Difficulty Defining Technical Data	235
6.6.5	Discussion	236
6.7	Conclusions	238
7	Conclusions and Future Work	239

7.1	Summary of Contributions of the Thesis	240
7.2	Publications	243
7.2.1	Conferences and Workshops	243
7.2.2	International Journals	244
7.2.3	Relevance of the Publications	245
7.3	Co-directed Projects	246
7.4	Future Work	247
7.4.1	Automated Production of Situational Methods	247
7.4.2	Method as a Service (MaaS)	250
7.4.3	Megamodeling in Method Engineering	251
7.4.4	Method Analysis and Monitoring	253
	Appendices	254
	A Comparative Analysis of SPEM 2.0 and BPMN 2.0	255
A.1	Method Modeling Criteria	256
A.2	Process Support Criteria	258
A.3	Conclusions	259
	B A Case Study: the gvMétrica Method	261
B.1	The gvMétrica Method	262
B.1.1	Design of Information Systems in gvMétrica	263
B.2	Applying the Methodological Approach	265
B.2.1	Method Design	266
B.2.1.1	Method Definition	266
B.2.1.2	Method Configuration	268
B.2.1.3	Executable Process Generation	270
B.2.2	Method Implementation	271
B.2.3	Method Execution	272
B.3	Conclusions	274

C Supplementary Material on the Evaluation Study	275
C.1 Characterization Form	275
C.2 User Acceptance Form	279
C.3 Interview Questions	279
C.4 Statistical Tests	282
C.4.1 Normality Tests	283
C.4.2 Non-parametric Tests	285
C.5 Coding Scheme	287
 Bibliography	 293