CONTENTS

I PRELIMINARIES 1

1 INTRODUCTION 3
- 1.1 Motivation 6
- 1.2 Problem Statement 8
- 1.3 Research Challenges 9
- 1.4 The Approach 11
- 1.5 Research Methodology 14
- 1.6 Thesis Context 16
- 1.7 Outline 16

2 BACKGROUND 19
- 2.1 Enablers of Our Approach 20
  - 2.1.1 Autonomic Computing 20
  - 2.1.2 Model-driven Engineering 25
  - 2.1.3 Models at Runtime 26
  - 2.1.4 Software Product Line Engineering 28
- 2.2 Target of Our Research 32
  - 2.2.1 Service Compositions 32
  - 2.2.2 Context 42
- 2.3 Conclusions 45

3 STATE OF THE ART 47
- 3.1 Research Areas 49
- 3.2 Taxonomy to Classify Research Works 51
  - 3.2.1 What Change is the Cause of Adaptation 53
  - 3.2.2 How the Service Composition Faces Changes 53
  - 3.2.3 When Changes are Carried Out 54
  - 3.2.4 Where Changes are Carried Out 55
  - 3.2.5 Maturity of the Approach 55
- 3.3 Related Work in the Main Research Areas 56
  - 3.3.1 Variability Constructs at the Language Level 58
  - 3.3.2 Brokers 61
  - 3.3.3 Models at Runtime 67
  - 3.3.4 DSPLs 74
- 3.4 Related Work in the Research Subareas 76
  - 3.4.1 Variability Modeling 77
  - 3.4.2 Uncertainty Management in the Open World 77

3.5 Conclusions 78
4 OVERVIEW OF THE APPROACH 83
4.1 Case Study 84
4.2 Main Building Blocks 87
4.2.1 Design-Related Building Blocks 88
4.2.2 Runtime-Related Building Blocks 89
4.3 A Framework for Autonomic Service Compositions 91
4.3.1 Design Phase 91
4.3.2 Dynamic Adaptation Phase 93
4.3.3 Dynamic Evolution Phase 94
4.4 Conclusions 95

II PAVING THE WAY FOR THE DYNAMIC ADJUSTMENT OF SERVICE COMPOSITIONS 97

5 MODEL-DRIVEN DESIGN FOR DYNAMIC ADAPTATION 99
5.1 Achieving Autonomic Service Compositions with MDE and DSPLE 100
5.2 A Process to Design Dynamic Adaptations 103
5.2.1 Create the Initial Composition Model 104
5.2.2 Create the Variability Model 105
5.2.3 Set Variability at the Composition Model 107
5.2.4 Create the Context Model 109
5.2.5 Define Context Conditions 111
5.2.6 Define Resolutions 112
5.2.7 Generate the Adaptation Space 114
5.2.8 Link Features to Service Operations 115
5.2.9 Verify Reconfigurations 117
5.2.10 Model-Driven Generation of WS-BPEL Code 124
5.3 Conclusions 131
6 MODELING TO FACE UNCERTAINTY IN THE OPEN WORLD 133
6.1 Dynamic Adaptation and Dynamic Evolution of Software 135
6.2 Conceptual Framework for Autonomic Service Compositions 137
6.3 Getting Ready to Face Uncertainty in the Open World 139
6.4 A Process to Design Dynamic Evolutions 141
6.5 Abstracting Requirements in the Requirements Model 141
6.6 Preserving the Requirements with Tactic Models 144
6.7 Reasoning about Unknown Context Events with Rule Premises 149

6.8 Conclusions 151

III DYNAMIC ADJUSTMENT OF SERVICE COMPOSITIONS 153

7 ACHIEVING DYNAMIC ADAPTATION THROUGH MODELS AT RUNTIME 155
7.1 MoRE-WS: An Extension of MoRE for Service Compositions 157
7.2 Monitoring the Context 158
7.3 Analyzing the Context 160
7.4 Planning the Adaptation 162
7.5 Executing the Adaptation 167
7.6 Conclusions 170

8 ACHIEVING DYNAMIC EVOLUTION THROUGH MODELS AT RUNTIME 173
8.1 Computing Infrastructure for Dynamic Evolutions 174
8.2 Planning the Evolution with the Evolution Planner 177
8.3 Using MoRE-WS for Dynamic Evolutions 179
8.4 Conclusions 185

IV APPLICABILITY AND PROVABILITY 187

9 TOOL SUPPORT AND METHOD CONTENTS 189
9.1 Plugable Method Contents 191
9.2 Designing Dynamic Adaptations 193
9.2.1 Create the Initial Composition Model 195
9.2.2 Create the Variability Model 196
9.2.3 Set Variability at the Composition Model 199
9.2.4 Create the Context Model 202
9.2.5 Define Context Conditions 203
9.2.6 Define Resolutions 203
9.2.7 Generate the Adaptation Space 204
9.2.8 Link Features to Service Operations 205
9.2.9 Verify Reconfigurations 206
9.2.10 Model-Driven Generation of WS-BPEL Code 208
9.3 Designing Dynamic Evolutions 208
9.3.1 Create the Requirements Model 209
9.3.2 Create the Tactic Models 210
9.3.3 Create the Weaving Model 211
9.3.4 Implement the Service Operations for Tactics 213
9.3.5 Create the Fragments of WS-BPEL Code to Invoke Tactics 214
9.3.6 Create the Rule Premises 214

9.4 MoRE-WS 215
9.5 Conclusions 219
10 VALIDATION 223
10.1 Evaluated Aspects 224
10.2 A Brief Introduction of the GQM Paradigm 227
10.3 Computing Infrastructure for the Experiments 227
10.4 Validation in the Design Phase 229
10.4.1 Generation Efficiency of Variability Model Configurations 229
10.4.2 Complexity Reduction of the Adaptation Space 230
10.4.3 Anomalies Reduction in the Variability Model and its Configurations 231
10.4.4 Verification Efficiency 232
10.5 Validation in the Dynamic Adaptation Phase 233
10.5.1 Context Observation Efficiency 234
10.5.2 Dynamic Adaptation Efficiency 236
10.5.3 Operability under Stress 240
10.6 Validation in the Dynamic Evolution Phase 242
10.6.1 Dynamic Evolution Efficiency 243
10.6.2 Inferences Accuracy 247
10.7 Conclusions 248

V CLOSING REMARKS 251
11 CONCLUSIONS AND FUTURE WORK 253
11.1 Contributions 254
11.2 Publications 258
11.3 International Collaboration 262
11.4 Codirected Master Thesis 262
11.5 Future Work 263
11.6 My Quest to Scientific Knowledge 265

VI APPENDICES 267
A IMPLEMENTATION DETAILS 269
A.1 Context Monitor Operations to Observe the Context 269
A.2 MoRE-WS Operations for Dynamic Adaptations 271
A.2.1 Analyze the Context 272
A.2.2 Planning the Adaptation 277
A.2.3 Executing the Adaptation 289
A.3 Operations for Dynamic Evolutions 293

A.3.1 Planning the Evolution with the Evolution Planner 294
A.3.2 Using MoRE-WS for Dynamic Evolutions 299
B CASE STUDY: BOOK SHOPPING PROCESS 307
B.1 Web Services 308
B.2 Context Conditions and Resolutions 309
B.3 WS-BPEL Template 312
B.4 Dynamic Adaptation Scenario 320
B.4.1 Requirements and Initial WS-BPEL Composition Schema 320
B.4.2 Dynamic Adaptation for B&NUnavailable 324
B.4.3 Dynamic Adaptation for UPSHiExecTime 331
B.5 Dynamic Evolution Scenario 339

BIBLIOGRAPHY 351