Contents

Part I The OO-Method and Software Production from Models

1 Let's Get Philosophical 3

2 The Purpose of this Work 7

3 The Need for New Development Environments 13
3.1 Software Development Today 14
3.1.1 Current Tools Are Inadequate 14
3.1.2 Development Processes Are Negatively Affected 15
3.2 Better Development Environments Are Needed 16

4 Object-Oriented Modelling as the Starting Point 19
4.1 The Chosen Model 19
4.2 Key Aspects of the Desired Method 21
4.2.1 Cataloguing of Conceptual Patterns 22
4.2.2 Enriching the Object Model 28
4.2.3 Software Architecture: Architectural Patterns 32

5 The OO-Method 39
5.1 Historical Background 39
5.2 General Features of the OO-Method 42
5.3 Other Approaches 43

Part II Conceptual Modelling: About the Problem Space

6 Conceptual Modelling Primitives 49
6.1 Conceptual Modelling. The Four Models 52

7 Object Model 55
7.1 Classes 55
7.1.1 Attributes 58
7.1.2 Services 62
7.1.3 Integrity Constraints 73
7.2 Relationships Between Classes 75
7.2.1 Agents and Agent Relationships 76
7.2.2 Association, Aggregation, and Composition 80
7.2.3 Specialisation 95
7.2.4 Parallel Composition 104
7.2.5 Visibility Between Related Classes 106
7.3 Complexity Management 111

8 Dynamic Model 115
8.1 State Transition Diagram 116
8.1.1 States 117
8.1.2 Transitions 118
8.1.3 Basic Statechart Diagram 120
8.1.4 Managing Complexity in State Transition Diagrams 121
8.1.5 Operations and State Transition Diagrams 122
8.1.6 Specialisation and State Transition Diagrams 123
8.2 Object Interaction Diagram 124
8.2.1 Triggers 125
8.2.2 Global Transactions and Global Operations 131

9 Functional Model 137
9.1 Event Effect Specification Using Evaluations 139
9.2 Evaluation Conditions 139
9.3 Categorization of Evaluations 141
9.4 Creation Events and the Functional Model 142
9.5 Destruction Events and the Functional Model 144
9.6 Specialisation and the Functional Model 144
9.7 Transactions, Operations, and the Functional Model 145

10 Presentation Model 147
10.1 General Structure 148
10.2 Basic Elements 151
10.2.1 Entry 151
10.2.2 Defined Selection 153
10.2.3 Argument Grouping 155
10.2.4 Argument Dependency 157
10.2.5 Filter 159
10.2.6 Order Criterion 163
10.2.7 Display Set 165
10.2.8 Navigation 167

10.2.9 Actions ... 169
10.3 Interaction Units ... 171
10.3.1 Service Interaction Units ... 172
10.3.2 Instance Interaction Unit ... 175
10.3.3 Population Interaction Unit ... 178
10.3.4 Master/Detail Interaction Unit ... 181
10.4 Action Hierarchy Tree ... 184
10.4.1 Agents and Action Hierarchy Trees ... 187
10.5 Constructing a Presentation Model ... 188

11 Conceptual Modelling of Legacy Systems ... 191
11.1 Introduction to the Concept of Legacy View ... 191
11.2 Object Model ... 192
11.2.1 Legacy Views ... 192
11.2.2 Relationships ... 201
11.3 Dynamic Model ... 207
11.3.1 State Transition Diagrams ... 207
11.3.2 Object Interaction Diagrams ... 208
11.4 Functional Model ... 208
11.5 Presentation Model ... 209
11.5.1 Basic Elements ... 209
11.5.2 Interaction Units ... 213
11.5.3 Action Hierarchy Tree ... 213

12 Conceptual Model Validation ... 215
12.1 Introduction ... 215
12.2 Object Model ... 217
12.2.1 Classes ... 217
12.2.2 Relationships Between Classes ... 221
12.3 Dynamic Model ... 223
12.3.1 State Transition Diagram ... 223
12.3.2 Object Interaction Diagram ... 224
12.4 Functional Model ... 224
12.4.1 Evaluations ... 224
12.5 Presentation Model ... 225
12.5.1 Basic Elements ... 225
12.5.2 Interaction Units ... 229
12.5.3 Action Hierarchy Tree ... 231

Part III Conceptual Model Compilation: from the Problem Space to the Solution Space

13 Transforming Models into Applications 235

14 Requirements for the Compilation of Conceptual Models 239

15 Application Execution Strategy 241
15.1 System Access 242
15.2 System View 242
15.3 System Interaction 243
15.3.1 Service Execution 244
15.3.2 Query Execution 255
15.3.3 Interaction Units 257

16 Application Architecture 261

17 Transformation Strategy 265
17.1 Mappings 265
17.2 Transformations 266

18 Building a Conceptual Model Compiler 269

19 Issues with Other Approaches 271
19.1 Issues with Modelling Languages 271
19.1.1 Lack of Adequacy 271
19.1.2 Imprecise Semantics 274
19.1.3 Strong Versus Weak Formalisms 274
19.1.4 Low Abstraction Level 275
19.1.5 Action Semantics 276
19.1.6 Formulae 277
19.1.7 Overspecification and Redundancy 277
19.2 Issues with Model Compilation 278
19.2.1 Lack of an Application Execution Strategy 278
19.2.2 Issues with Application Architectures 279
19.2.3 Insufficient Transformations 279
19.3 State of the Art 280

20 Analogies Between OO-Method and MDA 281
20.1 OO-Method and MDA 281
20.2 Properties of OO-Method Absent from MDA 282

21 The OO-Method Development Process 285

22 OO-Method Implementations 289
22.1 OLIVANOVA Modeler 289
22.2 OLIVANOVA Transformation Engines 290

Conclusions 293

References 295

Index 301