
CONTENTS

Preface xvii

1 Introduction

1.1	APPLICATION ORIENTATION – THE SUBJECT OF THIS BOOK	1
1.1.1	Motivation	1
1.1.2	Structure of This Book	3
1.2	THE TOOLS & MATERIALS APPROACH (T&M)	4
1.2.1	The T&M Approach in a Nutshell	4
1.2.2	T&M As a Method	8
1.3	PROJECTS BEHIND THIS BOOK	9
1.3.1	The Scope of the T&M Projects	10
1.4	THE EQUIPMENT MANAGEMENT SYSTEM EXAMPLE	14
1.5	REFERENCES	15

2 The T&M Object Metamodel

2.1	THE OBJECT METAMODEL	17
2.1.1	Introduction	17
2.1.2	Definition: The Object Metamodel	18
2.1.3	Context: What's the Purpose of an Object Metamodel?	18
2.1.4	Context: A Classification of Programming Languages	19
2.1.5	The Object Metamodel and the Software Model	20
2.1.6	Definition: Objects	20
2.1.7	Discussion: Object Identity	24
2.1.8	T&M Design: Structuring an Interface	25
2.1.9	Definition: Classes	25
2.1.10	Discussion: Generic Operations	27
2.1.11	T&M Design: Generic Operations	27
2.1.12	Discussion: The Object Life Cycle	28
2.1.13	T&M Design: The Object Life Cycle	29

2.1.14	Definition: Inheritance	30
2.1.15	Discussion: Inheritance	30
2.1.16	T&M Design: Inheritance	31
2.1.17	Discussion: Role Relationships as an <i>Alternative to Inheritance</i>	32
2.1.18	Definition: Use Relationships	32
2.1.19	Discussion: Use Relationships	33
2.1.20	Definition: Polymorphism	33
2.1.21	Definition: Abstract Classes	34
2.1.22	Discussion: Specification and Implementation	35
2.1.23	Definition: Loose Coupling	35
2.1.24	Discussion: Loose Coupling	36
2.2	MODULARIZATION	36
2.2.1	Introduction	36
2.2.2	Context: Modules and Object Orientation	36
2.2.3	Definition: Principles of Object-Oriented Modularization	37
2.3	THE CONTRACT MODEL	38
2.3.1	Introduction	38
2.3.2	Definition: Contract Model	38
2.3.3	Discussion: The Contract Model	39
2.3.4	Context: The Contract Model and Abstract Data Types	39
2.3.5	T&M Design: The Contract Model	40
2.4	TYPES	40
2.4.1	Introduction	40
2.4.2	Definition: Types	41
2.4.3	Context: The Theoretical Concept of Types	41
2.4.4	Discussion: Types	42
2.5	CLASSES AND TYPES	43
2.5.1	Introduction	43
2.5.2	Classes in Your Design	43
2.5.3	Differences between Types and Classes	43
2.5.4	Discussion: Classes versus Types	44
2.5.5	Background: Programming Languages and Types	44
2.5.6	T&M Design: Classes and Types	45
2.6	VALUES AND OBJECTS	45
2.6.1	Introduction	45
2.6.2	Characteristics of Values and Objects	46
2.6.3	Using Values	47
2.6.4	Context: Values and Objects in Programming Languages	48
2.6.5	Definition: Domain Values	49

2.6.6	T&M Design: Domain Values	49
2.6.7	<i>Implementing Domain Values</i>	49
2.7	METAOBJECT PROTOCOLS	50
2.7.1	Introduction	50
2.7.2	Motivation for a Metaobject Protocol	51
2.7.3	Definition: Metaobject Protocol (MOP)	51
2.7.4	Representing Your Application Model	51
2.7.5	Representing Your Runtime System	53
2.8	REFERENCES	55

3 Guiding Metaphors and Design Metaphors

3.1	INTRODUCTION	57
3.2	DESIGNING APPLICATION SOFTWARE	57
3.2.1	Definition: The Usage Model	58
3.2.2	Background: Methodology Books versus Design Guidelines	58
3.3	GUIDING METAPHORS FOR APPLICATION SOFTWARE	59
3.3.1	Background: Guiding Metaphors in Software Development	60
3.3.2	The “Object Worlds” Guiding Metaphor	60
3.3.3	The Direct Manipulation Guiding Metaphor	61
3.3.4	Discussion: Guiding Metaphor of Direct Manipulation	61
3.3.5	The Guiding Metaphor of the Factory	62
3.3.6	Discussion: The Factory Guiding Metaphor	62
3.4	DESIGN METAPHORS	63
3.4.1	Definition: Metaphors	63
3.5	T&M GUIDING METAPHORS AND DESIGN METAPHORS	64
3.5.1	A T&M Guiding Metaphor: The Expert Workplace	64
3.5.2	Background: The Supportive View	65
3.5.3	Discussion: Metaphors and Patterns	66
3.5.4	Definition: A Tool	67
3.5.5	<i>The Tool as a Design Metaphor</i>	67
3.5.6	T&M Design: Software Tools	67
3.5.7	Definition: Material	68
3.5.8	<i>Material as a Design Metaphor</i>	68
3.5.9	T&M Design: Software Materials	68
3.5.10	Definition: The Work Environment	69

3.5.11	The Work Environment as a Design Metaphor	69
3.5.12	T&M Design: The Work Environment	70
3.5.13	Definition: Automaton	70
3.5.14	The Automaton as a Design Metaphor	70
3.5.15	T&M Design: Software Automaton	71
3.5.16	Definition: The Container	71
3.5.17	The Container as a Design Metaphor	72
3.5.18	T&M Design: Containers	72
3.5.19	Discussion: Design Metaphors	73
3.6	WORKPLACE TYPES	73
3.6.1	Definition: Workplace Types	74
3.6.2	T&M Design: The Expert Workplace Type	74
3.6.3	T&M Design: The Functional Workplace Type	74
3.6.4	T&M Design: The Back-Office Workplace Type	75
3.6.5	T&M Design: The Electronic Commerce Frontend Workplace Type	76
3.7	REFERENCES	77
4	Patterns, Frameworks, and Components	
4.1	INTRODUCTION	79
4.2	BACKGROUND: PATTERNS, FRAMEWORKS, AND COMPONENTS	80
4.3	PATTERNS	80
4.3.1	Definition: Patterns	80
4.3.2	The Characteristics of a Pattern	81
4.3.3	A Taxonomy of T&M Patterns	83
4.3.4	Conceptual Patterns	83
4.3.5	Design Patterns	84
4.3.6	Programming Patterns	85
4.3.7	T&M Design: Design Patterns	86
4.3.8	T&M Design: Models, Metaphors, and Patterns	86
4.3.9	Background: Pattern Form	88
4.3.10	T&M Design: Pattern Form	88
4.3.11	Pattern Collections	89
4.4	FRAMEWORKS	90
4.4.1	Background: Class Libraries	90
4.4.2	Definition: Frameworks	91
4.4.3	Application Frameworks	91
4.4.4	Black-Box and White-Box Frameworks	92
4.4.5	Connectors between Frameworks	94
4.4.6	JWAM Framework: Layered Framework Architecture	95

4.5	COMPONENTS	96
4.5.1	Background: Software Components	96
4.5.2	Definition: Components	97
4.5.3	Current Component Products	98
4.5.4	Components and Frameworks	98
4.6	REFERENCES	99

5 Application-Oriented Software Development

5.1	INTRODUCTION	101
5.2	APPLICATION-ORIENTED SOFTWARE	101
5.2.1	Application Software	102
5.2.2	Definition: Application Orientation	102
5.2.3	Background: Application Orientation	103
5.2.4	Usage Quality	104
5.2.5	T&M Design: Structural Similarity	104
5.3	THE DEVELOPMENT PROCESS	105
5.3.1	Definition: Software Development	105
5.3.2	The Application-Oriented Development Process	105
5.3.3	Discussion: The Development Process	106
5.3.4	The Author-Critic Cycle	107
5.3.5	Discussion: The Author-Critic Cycle	107
5.3.6	Evolutionary System Development	108
5.3.7	Documentation in Software Development	108
5.3.8	Discussion: Documentation	109
5.3.9	Application-Oriented Development Documents	109
5.3.10	Discussion: Application-Oriented Document Types	110
5.3.11	T&M Design: Application-Oriented Document Types	111
5.3.12	Discussion: T&M Document Types	113
5.3.13	Project Documents	113
5.3.14	Documentation Guidelines	114
5.4	REFERENCES	115

6 Software Development as a Modeling Process

6.1	INTRODUCTION	117
6.2	A SIMPLIFIED SOFTWARE DEVELOPMENT MODEL	117
6.2.1	Discussion: A Descriptive Software Development Model	118
6.3	THE APPLICATION DOMAIN	119
6.3.1	Definition: Application Domain	119
6.3.2	Discussion: Analyzing the Application Domain	119

- 6.4 THE DOMAIN MODEL 120
 - 6.4.1 Modeling Your Application Domain 120
 - 6.4.2 Discussion: Modeling Your Application Domain 122
- 6.5 THE APPLICATION SYSTEM MODEL 122
 - 6.5.1 Context: The Application System Model 122
 - 6.5.2 Definition: Application System Model 123
 - 6.5.3 Discussion: Structural Similarity and Macrostructures 125
- 6.6 THE APPLICATION SYSTEM 125
 - 6.6.1 Definition: The Application System 126
 - 6.6.2 Discussion: The Application System 127
- 6.7 SOFTWARE DEVELOPMENT CONTEXTS 127
 - 6.7.1 Discussion: Software Development Contexts 127
 - 6.7.2 The Application Domain Context 127
 - 6.7.3 Discussion: The Application Domain Context 128
 - 6.7.4 Applied Technique 128
 - 6.7.5 Handling and Presentation 129
 - 6.7.6 Discussion: Handling and Presentation 130
- 6.8 CONTEXTS INFLUENCING THE SOFTWARE ARCHITECTURE 130
 - 6.8.1 Discussion: How Contexts Influence Your Software Architecture 130
- 6.9 REFERENCES 132

7 T&M Conceptual Patterns

- 7.1 CONCEPTUAL PATTERNS 135
 - 7.1.1 Conceptual Patterns in the Development Process 136
 - 7.1.2 The T&M Conceptual Patterns 136
- 7.2 A GUIDED TOUR OF THE T&M CONCEPTUAL PATTERNS 137
- 7.3 THE INTERRELATION OF TOOLS AND MATERIALS PATTERN 139
- 7.4 THE MATERIAL DESIGN PATTERN 142
- 7.5 THE TOOL DESIGN PATTERN 146
- 7.6 THE WORK ENVIRONMENT PATTERN 152
- 7.7 THE CONTAINER PATTERN 159
- 7.8 THE FORM PATTERN 164
- 7.9 THE AUTOMATON PATTERN 167
- 7.10 THE DOMAIN SERVICE PROVIDER PATTERN 171

7.11	THE TECHNICAL AUTOMATON PATTERN	176
7.12	THE PROBE PATTERN	179
7.13	THE ADJUSTING TOOL PATTERN	181

8 T&M Design Patterns

8.1	INTRODUCTION	185
8.2	A GUIDED TOUR OF THE T&M DESIGN PATTERNS	187
8.3	THE ASPECT PATTERN	190
8.3.1	Construction Part: Using Inheritance or Interfaces to Implement Aspects	195
8.3.2	Construction Part: Using Object Adapters to Implement Aspects	198
8.3.3	Construction Part: Using Development Tools to Realize Aspects	200
8.3.4	Construction Part: Alternatives to Using Aspects	201
8.4	THE SEPARATING FUNCTION AND INTERACTION PATTERN	202
8.5	THE TOOLS COMPOSITION PATTERN	204
8.5.1	Construction Part: Using Components to Build Tools	208
8.5.2	Construction Part: Using Components to Build Combination Tools	209
8.5.3	Construction Part: Identifying Tool Boundaries	210
8.6	THE FEEDBACK BETWEEN TOOL PARTS PATTERN	212
8.6.1	Construction Part: Event Pattern	214
8.6.2	Construction Part: Event Objects	215
8.6.3	Construction Part: Chain of Responsibility	217
8.6.4	Construction Part: Tool Component with Reaction Mechanisms	219
8.7	THE SEPARATING FP AND IP PATTERN	221
8.7.1	Construction Part: Interactive Part (IP)	225
8.7.2	Construction Part: FP	226
8.8	THE SEPARATING HANDLING AND PRESENTATION PATTERN	227
8.9	THE FEEDBACK BETWEEN INTERACTION FORMS AND IP PATTERN	232
8.10	THE DOMAIN VALUES PATTERN	236
8.10.1	Construction Part: Domain Value Classes	238
8.10.2	Construction Part: Immutable Domain Value Objects	240
8.10.3	Construction Part: Mutable Domain Value Objects	242

8.10.4	Construction Part: Implementing Domain Values as Streams	243
8.10.5	Construction Part: Domain Value Types by Configuration	243
8.11	THE DOMAIN CONTAINER PATTERN	244
8.11.1	Construction Part: Using Technical Containers to Implement Domain Containers	246
8.11.2	Construction Part: Loading Materials	247
8.11.3	Construction Part: Tables of Contents for Containers	248
8.11.4	Construction Part: Implementing Tables of Contents as Materials	249
8.11.5	Construction Part: Implementing Tables of Contents as Domain Values	249
8.11.6	Construction Part: Coping with Changes to Containers	250
8.12	THE FORM SYSTEM PATTERN	252
8.13	THE AUTOMATONS IN TECHNICALLY EMBEDDED SYSTEMS PATTERN	258
8.14	THE DOMAIN SERVICES PATTERN	264
8.15	THE ENVIRONMENT PATTERN	268
8.16	USING THE T&M DESIGN PATTERNS FOR THE JWAM FRAMEWORK	271
8.16.1	Materials Construction	271
8.16.2	Tools Construction	272
8.16.3	Domain Values	274
8.16.4	Presentation and Interaction Forms	278
8.16.5	Forms	278
8.16.6	Domain Services	279
8.16.7	Work Environment	279
8.17	REFERENCES	279

9 T&M Model Architecture

9.1	THE T&M MODEL ARCHITECTURE	281
9.2	THE DOMAIN CORE OF A SOFTWARE ARCHITECTURE	283
9.2.1	The Use Context	285
9.2.2	The Product Domain	287
9.2.3	The Business Domain	289
9.2.4	How Different Domains Relate	293
9.3	CONCEPTS AND ELEMENTS OF A T&M MODEL ARCHITECTURE	295

9.3.1	Components of an Object-Oriented Software Architecture	296
9.3.2	Elementary Rules for Combining Elements of a Software Architecture	296
9.3.3	Protocol-Based Layer Architectures	297
9.3.4	Object-Oriented Layer Architectures	299
9.3.5	The Layer Concept of the T&M Model Architecture	303
9.3.6	The Three-Tier Architecture	305
9.3.7	The T&M Model Architecture	307
9.4	DESIGN PATTERNS FOR THE T&M MODEL ARCHITECTURE	314
9.4.1	The Role Pattern	315
9.4.2	The Product Trader Pattern	327
9.5	REFERENCES	339

10 Supporting Cooperative Work

10.1	BACKGROUND: COMPUTER-SUPPORTED COOPERATIVE WORK	341
10.1.1	CSCW	341
10.1.2	Technical and Domain Transparency	344
10.2	IMPLICIT COOPERATION	345
10.3	EXPLICIT COOPERATION BY EXCHANGING MATERIALS	346
10.3.1	Cooperation Medium: Mailboxes	346
10.3.2	Cooperation Medium: Mailing System	349
10.4	EXPLICIT COOPERATION MODEL: TRANSACTION PROCESSING SUPPORT	350
10.4.1	The Concept of Process Patterns	353
10.4.2	Cooperation Medium: Routing Slips	354
10.5	REFERENCES	356

11 Interactive Application Systems and Persistence

11.1	BACKGROUND: INTERACTIVE APPLICATION SYSTEMS	357
11.2	PERSISTENCE SERVICES	358
11.2.1	Software Registry	358
11.2.2	The Basic Concept of a Software Registry	360
11.2.3	Cooperation Model for a Registry	361

11.2.4	Usage Model for a Registry	364
11.2.5	JWAM: Architecture for a Software Registry	365
11.2.6	The Generic Persistence Service	369
11.3	DESIGN CRITERIA TO IMPLEMENT PERSISTENCE	373
11.3.1	Client-Server Architecture	374
11.3.2	Identifiers	375
11.3.3	Technical Data Modeling and Structural Mapping	376
11.3.4	Querying and Data Warehousing	378
11.3.5	Load-on-Demand	379
11.3.6	Transactions and Locking	381
11.3.7	Class Evolution	382
11.3.8	Legacy Databases	382
11.4	REAL-WORLD EXAMPLES	383
11.4.1	JWAM Projects	383
11.4.2	MedIS and SyLab	386

12 The Development Process

12.1	BACKGROUND: EVOLUTIONARY AND TRADITIONAL PROCESS MODELS	393
12.1.1	The Context of Our Process Model	394
12.1.2	The Process Model's Application Orientation	394
12.1.3	The Classic Waterfall Model	395
12.1.4	The Spiral Model	397
12.1.5	An Idealized Evolutionary Process Model	398
12.2	TOPICS FOR A DEVELOPMENT STRATEGY	400
12.2.1	Sequence of Development Activities	400
12.2.2	Objectifying the Development Process	401
12.2.3	Lehman's Software Classification	402
12.2.4	The Cooperative Development Process	405
12.2.5	Organizational and Domain Integration	406
12.2.6	Developing an IT Organization	409
12.3	QUALITY ASSURANCE IN THE DEVELOPMENT PROCESS	413
12.3.1	Direct User Integration	413
12.3.2	Prototyping	414
12.3.3	Reviews	414
12.3.4	Pair Programming	416
12.3.5	Refactoring	418
12.4	QUALITY ASSURANCE IN CONSTRUCTION	419
12.4.1	Characteristics of OO Testing	419
12.4.2	Testing OO Programs	421

12.5	PROJECT MANAGEMENT	425
12.5.1	Fundamental Activities of the Management Process	425
12.5.2	The Contexts of the Management Process	427
12.6	PROJECT PLANNING CONCEPTS AND TECHNIQUES	428
12.6.1	Project Calibration	428
12.6.2	Project Goals	430
12.6.3	Decision Principles	431
12.6.4	Project Establishment	432
12.7	STRUCTURING A PROJECT BY SYSTEM DECOMPOSITION	432
12.7.1	Core System and Special-Purpose Systems	432
12.7.2	Core System and Extension Levels	434
12.8	SCHEDULING AND TASK PLANNING	437
12.8.1	General Rules for Time Estimates	437
12.8.2	Planning the Project Stages	438
12.8.3	Using Base Lines for Detailed Planning	441
12.8.4	The UP and T&M Project Planning	443
12.9	DISCUSSING T&M, UNIFIED PROCESS, AND XP	451
12.9.1	Structure of the UP and T&M Development Processes	451
12.10	REFERENCES	456

13 T&M Document Types

13.1	SCENARIOS	459
13.1.1	Using Scenarios in the Development Process	462
13.1.2	Subtypes of Scenarios	464
13.1.3	Scenarios and UML	466
13.2	INTERVIEWS	467
13.2.1	The Interview Process	468
13.3	THE CONCEPT MODEL	472
13.3.1	Using a Concept Model in the Development Process	475
13.3.2	Concept Models and UML	475
13.4	GLOSSARIES	475
13.4.1	Using a Glossary in the Development Process	477
13.4.2	Glossaries, UML, and UP	479
13.5	SYSTEM VISIONS	479
13.5.1	Using System Visions in the Development Process	480
13.5.2	System Visions and XP	482
13.5.3	System Visions and UML	482

13.6	PROTOTYPES	483	
13.6.1	Using Prototypes in the Development Process	485	
13.6.2	Prototypes and UP	487	
13.7	COOPERATION PICTURES	488	
13.7.1	Cooperation Pictures in the Development Process	490	
13.7.2	Cooperation Pictures and UML	494	
13.8	PURPOSE TABLES	494	
13.8.1	Using Purpose Tables in the Development Process	495	
13.8.2	Purpose Tables and UML	496	
13.9	TECHNICAL DOCUMENT TYPES IN UML	496	
13.9.1	Class Diagrams	497	
13.9.2	Object Diagrams	497	
13.9.3	Use Case Diagrams	498	
13.9.4	Interaction Diagrams	498	
13.9.5	Statechart Diagrams	498	
13.9.6	Activity Diagrams	498	
13.9.7	Component Diagrams	498	
13.9.8	Deployment Diagrams	498	
13.9.9	Application-Oriented and Technical Documents	499	
13.10	REFERENCES	499	

Index 501