Contents

About the Author ... xii

About the Development Editor ... xiii

About the Technical Editor ... xiii

Acknowledgments ... xv

Foreword ... xvii

Introduction ... xix

Chapter 1 The Starting Point ... 1

Knowing Your Product—Software Defined ... 2
Overcoming the Application-Centric View ... 3
- Desperately Seeking Reusability ... 5
- The (Im)practicality of High Reuse ... 5
- Harsh Realities of Application-Centric Reuse ... 6

Revealing the Illusion of Rapid Application Development ... 8
Embracing the Architecture-Centric View ... 10
Exploring Pattern-Centric Development ... 13
Using Metamorphism versus Polymorphism ... 16
Understanding the Language of Symbols ... 20
- Patterns in Mathematics ... 20
- Abstracting Knowledge ... 22
- Dynamic versus Static Representation ... 24

Realizing How Semantics Activate Knowledge ... 26
- Developer in a Box ... 28
- Development Projects ... 28
- Smooth Delivery Effect ... 30
- Implications to Development Teams ... 32
- Implications on Revenue ... 33
- Keeping the Momentum ... 34

Project 1—Groundwork ... 34
- TurboCollection ... 35
- Property Manipulation ... 39
- Factories and Object Recycle ... 42

Chapter 2 Building Blocks 47

Understanding the Common Application Architecture 48
The Interface Rules 51
- Building Application-Independent Reusable Objects 53
- Understanding Interface Dynamics 57

Examining Object Interaction 62
- Early Binding 63
- Late Binding 63
- Reuse versus Accelerated Reuse 66

Interpreting Structural Patterns 66
- The Singleton Pattern 67
- The Adapter pattern (a.k.a. Wrapper) 68
- The Flyweight Pattern 72

Implementing Stabilization Rules with SOAL 73
- Understanding the Interface Pattern 75
- SOAL, Take Two 91

Managing Resources 92
- The Factory 93
- Abstract Factories 93
- The Prototype 104

Comparing Application and Metamorphic Models 105
Project 2—Structures 108
- StringAdapter 108
- Text File Manipulation and Interpretation 110

Chapter 3 Best Behavior 115

Enforcing Behavioral Integrity 116
Using Metadata as Fuel 118
Leashing the Development Process 120
- Tell the Truth about User Requirements 121
- Understanding How Change Affects Product Behavior 124
- Applying an Appropriate Methodology 125
- Stalking Rogues on Teams 126
- Implementing Externally Programmable Behavior 128

Utilizing Behavioral Control Models 130
- The Flow-Based, Process-Driven Model 130
- The Event-Based, Interrupt-Driven Model 131
- Software Control 131

Exploring Behavioral Patterns 132
- The Iterator 132
- The Interpreter 133

The Observer/Visitor *137*
The Template *140*
Frameworks *144*
Simple Syntax *146*
Achieving Pattern-Based Behavioral Control *148*
Reality #1: Nothing "Real" Exists Inside the Computer's Run-Time Space *149*
Reality #2: Use Software to Transform the Statically Referenced Entities into Dynamic Entities *149*
Reality #3: The Entities and Their Behaviors Are Governed by Rules. The Default Rules (Software Code) Are Static *149*
Reality #4: Through Interpreter Patterns, We Can Use Software to Transform the Normally Static Governing Rules into Dynamic Rules *150*
Reality #5: Dynamically Manufacture the End-User Experience *150*
Reality #6: The Application Is External to the Software *151*
Building Valuable and Practical Components *152*
Obtaining Expert Advice *154*
Applying Behaviors *157*
Enterprise Application Integration (EAI) *158*
What Is the Real Value? *161*
Project 3—Behaviors *162*
Observers and Automatic Behavior Control *163*
Conditional Operators *167*
Looking Deeper *173*

Chapter 4 Power Tools *175*

Organizing, Interpreting, and Activating the Metamorphic Model *175*
Examining Structural Concepts *176*
Ode to the Node *178*
The ParmNode Flyweight Container *180*
Microsoft's Simple Collection Iterator *181*
The TurboCollection Iterator *182*
TurboCollection as a Binary Tree *188*
TurboCollection Interoperability *190*
Understanding Multiple Dimensions *191*
Mechanics of Iteration *193*
StringAdapter for Interpeter Support *195*
mathAdapter *198*
TextIO as Text Stream Manager *207*
Container as Row-Column Data Cache *210*
Power Tool Checkpoint *220*

Building a Programmable Application Prototype 221
Structural Reference 221
Interface Patterns 222
Structural and Behavioral Metadata 224
Structural Metadata Categories 225
Behavioral Metadata Categories 230
Application Prototype Checkpoint 243
Project 4—Interpreters 244
Interpretation Mechanics 244
But Wait, There's More 268

Chapter 5 Virtual Frameworks—Error Control 273

Leashing the Computing Environment 275
Portability and Compatibility 275
System Classes 276
Controlling Startup and Initialization 277
Controlling the Chaos 279
Bugs versus Errors 279
Slaying Bugs with *El Jefe De Insecto*s 280
Reducing Error Recovery Options 281
Chaos Concepts 282
Implementing Chaos Rules 287
Rule #1: Standardize, and Do Not Deviate from "Valid Result Values" 288
Rule #2: Log It—If It's Not Recorded, It Never Happened 290
Rule #3: Recover at the Source, *Not* the Consumer 293
Rule #4: Never Raise Error Traps 294
Rule #5: Provide for a Run-Time Log 295
Project Five—Error Control 298

Chapter 6 Virtual Frameworks—Modeling Data Dynamically 305

Developing the Data Model 306
Managing Changes to the Data Model 307
Realizing the Benefits of a Dynamic Model 309
Performing Dynamic Aliasing 316
Synonym 317
Surrogate 317
Pseudonym 318
Automatic Behavior 318

Implementing the vDDM Adapters 323
The dbEngine Adapter 324
The dbBroker Adapter 324
The dbData Adapter 325
The dbInterpreter 325
The dbTemplate Class 325
Interpreter Mechanics 326
Leashing the Power of Dynamic Integration 337
vSQL Configuration Management 338
Wrapping the Engine Itself 339
Refining Data-Awareness 341
Exporting Application Logic 342
Stored Procedures 342
Integrity Considerations 342
Impact Review 343
Project 6—Data Integration 344
Iterators and Container Patterns 345
Sorting 348
Container and the Interpreter Mechanics 351
Performance Configuration Strategies 355
Database Management 357
Examining Synonym Behavior 360
Examining vSQL Behavior 362
Examining vDDM Run-Time Structural Contents 365
Generating Data 366

Chapter 7 Virtual Frameworks—Screen Prototyping 373

Leashing Screen and Report Layouts 374
Driving Screens and Reports 374
Banging Out the Screens 375
Creating Dynamic Forms 378
Controlling Forms 382
Screen Prototyping Summary 393
Summarizing Application Prototypes, vDDM, and vUIM 395
Project 7—Fun with XML 398
Epilogue 405
Requirements Drive It All 406
See You at Virtual Mach! 408

Notes 411

Index 413