

<<企业应用架构模式>>

图书基本信息

书名：<<企业应用架构模式>>

13位ISBN编号：9787115204226

10位ISBN编号：7115204225

出版时间：2009

出版时间：人民邮电出版社

作者：Martin Fowler

页数：533

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<企业应用架构模式>>

### 前言

1999年春天，我飞抵芝加哥为ThoughtWorks公司正在开发的一个项目担任顾问。

ThoughtWorks是一个规模虽小但正在快速成长的应用开发公司。

这个项目属于那种雄心勃勃的企业级应用——一个后端租赁系统。

简单说，它处理的是租户签字认可后所有与租赁有关的事务，包括发送账单、处理某些人对所租房产的改造、追踪那些未按时付款的租户、处理提前退租的情况等。

这听起来好像并不太难实现，但是你大概没有意识到，租赁合同极度复杂并且总在不断变化。

它的业务“逻辑”几乎不能套用任何已有的逻辑模式，因为那些“逻辑”归根到底是商人们为了抢生意而制定的，一些奇怪的小改动都可能对赢得某笔生意起关键作用。

而每次这样的成功就意味着系统复杂性的增加。

我感兴趣的正是这类问题：如何捕获这些复杂情况，并设计一个面向对象的系统使问题更容易解决。

事实上，我认为面向对象的最大优点就在于它能够使复杂逻辑易于处理。

为复杂业务问题开发一个好的领域模型（Domain Model，116）很困难，但结果往往非常令人满意。

但问题还没有完。

我们的领域模型必须持久存储到数据库中，与许多项目一样，我们使用的也是关系数据库。

我们还必须将该模型与用户界面连接起来，还要支持远程应用程序使用本软件，并将软件与第三方的软件包集成。

所有这些工作都基于一种称作J2EE的新技术，当时世界上没有人在这方面具备真正的实战经验。

## <<企业应用架构模式>>

### 内容概要

面向对象大师Martin Fowler及其专家级合作者将40多种常用解决方案转化成模式，为我们提供了这本能够应用于任何一种企业应用平台的、关于解决方案的参考书。

《企业应用架构模式(英文版)》叙述深入浅出，采用大量UML图进一步阐明有关概念。前面介绍企业应用的背景知识，如分层架构、Web表现、业务逻辑、数据库映射、并发、会话、分布策略等。

在此基础上，随后的各章分别对与这些背景知识相关的设计模式进行了详细的介绍，并配以详细的Java代码或C#代码示例。

《企业应用架构模式(英文版)》适合设计和构建企业应用的软件架构师、设计人员和编程人员阅读，同时也可作为高等院校计算机专业及软件学院相关课程的参考教材。

## <<企业应用架构模式>>

### 作者简介

作者：(美国)Martin FowlerMartin Fowler，享誉世界的软件开发大师。

现为著名软件开发咨询公司ThoughtWorks的首席科学家。

他在面向对象分析与设计、UML、设计模式、软件开发方法学、XP、重构等方面都有重要贡献。

他是全球最具影响力的技术作家之一，除本书外，他的《分析模式》、《UML精粹》和《重构》等著作都已经成为经典。

## 书籍目录

Preface xvii Who This Book Is For xx Acknowledgments xxi Colophon xxiii Introduction 1 Architecture 1  
 Enterprise Applications 2 Kinds of Enterprise Application 5 Thinking About Performance 6 Patterns 9 The Structure  
 of the Patterns 11 Limitations of These Patterns 13 PART 1: The Narratives 15 Chapter 1: Layering 17 The  
 Evolution of Layers in Enterprise Applications 18 The Three Principal Layers 19 Choosing Where to Run Your  
 Layers 22 Chapter 2: Organizing Domain Logic 25 Making A Choice 29 Service Layer 30 Chapter 3: Mapping to  
 Relational Databases 33 Architectural Patterns 33 The Behavioral Problem 38 Reading in Data 40 Structural  
 Mapping Patterns 41 Mapping Relationships 41 Inheritance 45 Building the Mapping 47 Double Mapping 48  
 Using Metadata 49 Database Connections 50 Some Miscellaneous Points 52 Further Reading 53 Chapter 4: Web  
 Presentation 55 View Patterns 58 Input Controller Patterns 61 Further Reading 61 Chapter 5: Concurrency (by  
 Martin Fowler And David Rice) 63 Concurrency Problems 64 Execution Contexts 65 Isolation And Immutability  
 66 Optimistic And Pessimistic Concurrency Control 67 Preventing Inconsistent Reads 68 Deadlocks 70  
 Transactions 71 ACID 71 Transactional Resources 72 Reducing Transaction Isolation for Liveness 73 Business And  
 System Transactions 74 Patterns for Offline Concurrency Control 76 Application Server Concurrency 78 Further  
 Reading 80 Chapter 6: Session State 81 The Value of Statelessness 81 Session State 83 Ways to Store Session State 84  
 Chapter 7: Distribution Strategies 87 The Allure of Distributed Objects 87 Remote And Local Interfaces 88 Where  
 You Have to Distribute 90 Working with the Distribution Boundary 91 Interfaces for Distribution 92 Chapter 8:  
 Putting It All Together 95 Starting with the Domain Layer 96 Down to the Data Source Layer 97 Data Source for  
 Transaction Script (11 O) 97 Data Source for Table Module (125) 98 Data Source for Domain Model (116) 98 The  
 Presentation Layer 99 Some Technology-Specific Advice 100 Java And J2EE 100 NET 101 Stored Procedures 102  
 Web Services 103 Other Layering Schemes 103 PART 2: The Patterns 107 Chapter 9: Domain Logic Patterns 109  
 Transaction Script 110 How It Works 110 When to Use It 111 The Revenue Recognition Problem 112 Example:  
 Revenue Recognition (Java) 113 Domain Model 116 How It Works 116 When to Use It 119 Further Reading 119  
 Example: Revenue Recognition (Java) 120 Table Module 125 How It Works 126 When to Use It 128 Example:  
 Revenue Recognition with A Table Module (C#) 129 Service Layer (by Randy Stafford) 133 How It Works 134  
 When to Use It 137 Further Reading 137 Example: Revenue Recognition (Java) 138 Chapter 10: Data  
 Source Architectural Patterns 143 Table Data Gateway 144 How It Works 144 When to Use It 145 Further Reading  
 146 Example: Person Gateway (C#) 146 Example: Using ADO NET Data Sets (C#) 148 Row Data Gateway 152  
 How It Works 152 When to Use It 153 Example: A Person Record (Java) 155 Example: A Data Holder for A  
 Domain Object (Java) 158 Active Record 160 How It Works 160 When to Use It 161 Example: A Simple Person  
 (Java) 162 Data Mapper 165 How It Works 165 When to Use It 170 Example: A Simple Database Mapper (Java)  
 171 Example: Separating the Finders (Java) 176 Example: Creating An Empty Object (Java) 179 Chapter 11:  
 Object-Relational Behavioral Patterns 183 Unit of Work 184 How It Works 184 When to Use It 189 Example: Unit  
 of Work with Object Registration (Java) (by David Rice) 190 Identity Map 195 How It Works 195 When to Use It  
 198 Example: Methods for An Identity Map (Java) 198 Lazy Load 200 How It Works 200 When to Use It 203  
 Example: Lazy Initialization (Java) 203 Example: Virtual Proxy (Java) 203 Example: Using A Value Holder (Java)  
 205 Example: Using Ghosts (C#) 206 Chapter 12: Object-Relational Structural Patterns 215 Identity Field 216 How  
 It Works 216 When to Use It 220 Further Reading 221 Example: Integral Key (C#) 221 Example: Using A Key  
 Table (Java) 222 Example: Using A Compound Key (Java) 224 Foreign Key Mapping 236 How It Works 236  
 When to Use It 239 Example: Single-Valued Reference (Java) 240 Example: Multitable Find (Java) 243 Example:  
 Collection of References (C#) 244 Association Table Mapping 248 How It Works 248 When to Use It 249  
 Example: Employees And Skills (C#) 250 Example: Using Direct SQL (Java) 253 Example: Using A Single Query  
 for Multiple Employees (Java) (by Matt Foemmel And Martin Fowler) 256 Dependent Mapping 262 How It  
 Works 262 When to Use It 263 Example: Albums And Tracks (Java) .. 264 Embedded Value 268 How It Works 268  
 When to Use It 268 Further Reading 270 Example: Simple Value Object (Java) 270 Serialized LOB 272 How It  
 Works 272 When to Use It 274 Example: Serializing A Department Hierarchy in XML (Java) 274 Single Table

<<企业应用架构模式>>

Inheritance 278 How It Works 278 When to Use It 279 Example:A Single Table for Players (C#) 280 LoadingAn Object from the Database 281 Class Table Inheritance 285 How It Works 285 When to Use It 286 Further Reading 287 Example: PlayersAnd Their Kin (C#) 287 Concrete Table Inheritance 293 How It Works 293 When to Use It 295 Example: Concrete Players (C#) 296 Inheritance Mappers 302 How It Works 303 When to Use It 304 Chapter 13: Object-Relational Metadata Mapping Patterns 305 Chapter 14: Web Presentation Patterns 329 Chapter 15: Distribution Patterns 387 Chapter 16: Offline Concurrency Patterns 415 Chapter 17: Session State Patterns 455 Chapter 18: Base Patterns 465 Index 517

## <<企业应用架构模式>>

### 章节摘录

插图：The benefit of Service Layer is that it defines a common set of application operations available to many kinds of clients and it coordinates an application's response in each operation. The response may involve application logic that needs to be transacted atomically across multiple transactional resources. Thus, in an application with more than one kind of client of its business logic, and complex responses in its use cases involving multiple transactional resources, it makes a lot of sense to include a Service Layer with container-managed transactions, even in an undistributed architecture. The easier question to answer is probably when not to use it. You probably don't need a Service Layer if your application's business logic will only have one kind of client—say, a user interface—and its use case responses don't involve multiple transactional resources. In this case your Page Controllers can manually control transactions and coordinate whatever response is required, perhaps delegating directly to the Data Source layer. But as soon as you envision a second kind of client, or a second transactional resource in use case responses, it pays to design in a Service Layer from the beginning.

## <<企业应用架构模式>>

### 编辑推荐

《企业应用架构模式(英文版)》是软件开发大师Martin Fowler的代表作。

采用模式的形式系统总结了业界多年积累的经验，被称为企业级应用开发领域的“圣经”，出版以来一直畅销不衰，至今仍然无可替代。

作者在精彩地阐述了企业应用开发设计中的核心原则基础上。

详细、生动地讲述了51个模式并给出主流平台（Java和.NET）中的应用实例，更分析了许多相似模式之间的差异。

提供了具体运用和选择这些模式的大量经验之谈，使你不仅知其然。

更知其所以。

这是一部软件开发领域不朽的经典。

任何一位真正的软件开发人员都不可错过。

随着信息技术的广泛应用。

系统需要处理的数据量越来越大，企业级软件开发已经渐成主流，而开发人员面临的困难与挑战也是显而易见的。

更糟糕的是，这一领域的资料一直非常缺乏。



<<企业应用架构模式>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>