

《领域驱动设计》

图书基本信息

书名：《领域驱动设计》

13位ISBN编号：9787115376751

出版时间：2016-6-1

作者：[美] Eric Evans

页数：370

译者：赵俐,盛海艳,刘霞

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《领域驱动设计》

内容概要

本书是领域驱动设计方面的经典之作，修订版更是对之前出版的中文版进行了全面的修订和完善。全书围绕着设计和开发实践，结合若干真实的项目案例，向读者阐述如何在真实的软件开发中应用领域驱动设计。书中给出了领域驱动设计的系统化方法，并将人们普遍接受的一些实践综合到一起，融入了作者的见解和经验，展现了一些可扩展的设计新实践、已验证过的技术以及便于应对复杂领域的软件项目开发的基本原则。

《领域驱动设计》

作者简介

Eric Evans “领域驱动设计之父”，世界杰出软件建模专家。他创建了Domain Language公司，致力于帮助公司机构创建与业务紧密相关的软件。他在世界各地宣讲领域驱动设计（Domain-Driven Design，DDD）的思想，开设课程，参加会议，接受专访，拥有大批的追随者。从20世纪80年代开始，他就以设计师和程序员的双重身份参与过许多大型面向对象系统的设计和开发，涉及各种复杂的业务和技术领域。同时，他还培训和指导过许多开发团队开展极限编程实践。

书籍目录

目录

第一部分 运用领域模型

第1章 消化知识 5

1.1 有效建模的要素 9

1.2 知识消化 10

1.3 持续学习 11

1.4 知识丰富的设计 12

1.5 深层模型 15

第2章 交流与语言的使用 16

2.1 模式：UBIQUITOUS LANGUAGE 16

2.2 “大声地”建模 21

2.3 一个团队，一种语言 22

2.4 文档和图 24

2.4.1 书面设计文档 25

2.4.2 完全依赖可执行代码的情况 27

2.5 解释性模型 27

第3章 绑定模型和实现 29

3.1 模式：MODEL-DRIVEN DESIGN 30

3.2 建模范式和工具支持 32

3.3 揭示主旨：为什么模型对用户至关重要 38

3.4 模式：HANDS-ON MODELER 39

第二部分 模型驱动设计的构造块

第4章 分离领域 43

4.1 模式：LAYERED ARCHITECTURE 43

4.1.1 将各层关联起来 46

4.1.2 架构框架 47

4.2 领域层是模型的精髓 48

4.3 模式：THE SMART UI “反模式” 48

4.4 其他分离方式 50

第5章 软件中所表示的模型 51

5.1 关联 52

5.2 模式：ENTITY（又称为REFERENCE OBJECT） 56

5.2.1 ENTITY建模 59

5.2.2 设计标识操作 60

5.3 模式：VALUE OBJECT 62

5.3.1 设计VALUE OBJECT 64

5.3.2 设计包含VALUE OBJECT的关联 67

5.4 模式：SERVICE 67

5.4.1 SERVICE与孤立的领域层 69

5.4.2 粒度 70

5.4.3 对SERVICE的访问 70

5.5 模式：MODULE（也称为PACKAGE） 71

5.5.1 敏捷的MODULE 72

5.5.2 通过基础设施打包时存在的隐患 73

5.6 建模范式 75

5.6.1 对象范式流行的原因 76

5.6.2 对象世界中的非对象 77

- 5.6.3 在混合范式中坚持使用MODEL-DRIVEN DESIGN 78
- 第6章 领域对象的生命周期 80
 - 6.1 模式：AGGREGATE 81
 - 6.2 模式：FACTORY 89
 - 6.2.1 选择FACTORY及其应用位置 91
 - 6.2.2 有些情况下只需使用构造函数 93
 - 6.2.3 接口的设计 94
 - 6.2.4 固定规则的相关逻辑应放置在哪里 94
 - 6.2.5 ENTITY FACTORY与VALUE OBJECT FACTORY 95
 - 6.2.6 重建已存储的对象 95
 - 6.3 模式：REPOSITORY 97
 - 6.3.1 REPOSITORY的查询 101
 - 6.3.2 客户代码可以忽略REPOSITORY的实现，但开发人员不能忽略 102
 - 6.3.3 REPOSITORY的实现 103
 - 6.3.4 在框架内工作 104
 - 6.3.5 REPOSITORY与FACTORY的关系 104
 - 6.4 为关系数据库设计对象 106
- 第7章 使用语言：一个扩展的示例 108
 - 7.1 货物运输系统简介 108
 - 7.2 隔离领域：引入应用层 110
 - 7.3 将ENTITY和VALUE OBJECT区别开 110
 - 7.4 设计运输领域中的关联 112
 - 7.5 AGGREGATE边界 113
 - 7.6 选择REPOSITORY 113
 - 7.7 场景走查 115
 - 7.7.1 应用程序特性举例：更改Cargo的目的地 115
 - 7.7.2 应用程序特性举例：重复业务 116
 - 7.8 对象的创建 116
 - 7.8.1 Cargo的FACTORY和构造函数 116
 - 7.8.2 添加Handling Event 117
 - 7.9 停一下，重构：Cargo AGGREGATE 的另一种设计 118
 - 7.10 运输模型中的MODULE 120
 - 7.11 引入新特性：配额检查 122
 - 7.11.1 连接两个系统 123
 - 7.11.2 进一步完善模型：划分业务 124
 - 7.11.3 性能优化 125
 - 7.12 小结 126
- 第三部分 通过重构来加深理解
- 第8章 突破 131
 - 8.1 一个关于突破的故事 131
 - 8.1.1 华而不实的模型 132
 - 8.1.2 突破 133
 - 8.1.3 更深层模型 135
 - 8.1.4 冷静决策 137
 - 8.1.5 成果 138
 - 8.2 机遇 138
 - 8.3 关注根本 138
 - 8.4 后记：越来越多的新理解 139
- 第9章 将隐式概念转变为显式概念 140

- 9.1 概念挖掘 140
 - 9.1.1 倾听语言 140
 - 9.1.2 检查不足之处 144
 - 9.1.3 思考矛盾之处 148
 - 9.1.4 查阅书籍 148
 - 9.1.5 尝试，再尝试 150
- 9.2 如何为那些不太明显的概念建模 150
 - 9.2.1 显式的约束 151
 - 9.2.2 将过程建模为领域对象 153
 - 9.2.3 模式：SPECIFICATION 154
 - 9.2.4 SPECIFICATION的应用和实现 156
- 第10章 柔性设计 168
 - 10.1 模式：INTENTION-REVEALING INTERFACES 169
 - 10.2 模式：SIDE-EFFECT-FREE FUNCTION 173
 - 10.3 模式：ASSERTION 177
 - 10.4 模式：CONCEPTUAL CONTOUR 181
 - 10.5 模式：STANDALONE CLASS 184
 - 10.6 模式：CLOSURE OF OPERATION 186
 - 10.7 声明式设计 188
 - 10.8 声明式设计风格 190
 - 10.9 切入问题的角度 197
 - 10.9.1 分割子领域 197
 - 10.9.2 尽可能利用已有的形式 198
- 第11章 应用分析模式 206
- 第12章 将设计模式应用于模型 217
 - 12.1 模式：STRATEGY（也称为POLICY） 218
 - 12.2 模式：COMPOSITE 221
 - 12.3 为什么没有介绍FLYWEIGHT 226
- 第13章 通过重构得到更深层的理解 227
 - 13.1 开始重构 227
 - 13.2 探索团队 227
 - 13.3 借鉴先前的经验 228
 - 13.4 针对开发人员的设计 229
 - 13.5 重构的时机 229
 - 13.6 危机就是机遇 230
- 第四部分 战略设计
 - 第14章 保持模型的完整性 233
 - 14.1 模式：BOUNDED CONTEXT 235
 - 14.2 模式：CONTINUOUS INTEGRATION 239
 - 14.3 模式：CONTEXT MAP 241
 - 14.3.1 测试CONTEXT的边界 247
 - 14.3.2 CONTEXT MAP的组织 and 文档化 247
 - 14.4 BOUNDED CONTEXT之间的关系 248
 - 14.5 模式：SHARED KERNEL 248
 - 14.6 模式：CUSTOMER/SUPPLIER DEVELOPMENT TEAM 250
 - 14.7 模式：CONFORMIST 253
 - 14.8 模式：ANTICORRUPTION LAYER 255
 - 14.8.1 设计ANTICORRUPTION LAYER的接口 256

- 14.8.2 实现ANTICORRUPTION LAYER 256
- 14.8.3 一个关于防御的故事 259
- 14.9 模式：SEPARATE WAY 260
- 14.10 模式：OPEN HOST SERVICE 261
- 14.11 模式：PUBLISHED LANGUAGE 262
- 14.12 “大象”的统一 264
- 14.13 选择你的模型上下文策略 267
 - 14.13.1 团队决策或更高层决策 268
 - 14.13.2 置身上下文中 268
 - 14.13.3 转换边界 268
 - 14.13.4 接受那些我们无法更改的事物：描述外部系统 269
 - 14.13.5 与外部系统的关系 269
 - 14.13.6 设计中的系统 270
 - 14.13.7 用不同模型满足特殊需要 270
 - 14.13.8 部署 271
 - 14.13.9 权衡 271
 - 14.13.10 当项目正在进行时 272
- 14.14 转换 272
 - 14.14.1 合并CONTEXT：SEPARATE WAY SHARED KERNEL 273
 - 14.14.2 合并CONTEXT：SHARED KERNEL CONTINUOUS INTEGRATION 274
 - 14.14.3 逐步淘汰遗留系统 275
 - 14.14.4 OPEN HOST SERVICE PUBLISHED LANGUAGE 276
- 第15章 精炼 277
 - 15.1 模式：CORE DOMAIN 278
 - 15.1.1 选择核心 280
 - 15.1.2 工作的分配 280
 - 15.2 精炼的逐步提升 281
 - 15.3 模式：GENERIC SUBDOMAIN 282
 - 15.3.1 通用不等于可重用 286
 - 15.3.2 项目风险管理 287
 - 15.4 模式：DOMAIN VISION STATEMENT 287
 - 15.5 模式：HIGHLIGHTED CORE 289
 - 15.5.1 精炼文档 289
 - 15.5.2 标明CORE 290
 - 15.5.3 把精炼文档作为过程工具 291
 - 15.6 模式：COHESIVE MECHANISM 292
 - 15.6.1 GENERIC SUBDOMAIN与COHESIVE MECHANISM的比较 293
 - 15.6.2 MECHANISM是CORE DOMAIN一部分 294
 - 15.7 通过精炼得到声明式风格 294
 - 15.8 模式：SEGREGATED CORE 295
 - 15.8.1 创建SEGREGATED CORE的代价 296
 - 15.8.2 不断发展演变的团队决策 296
 - 15.9 模式：ABSTRACT CORE 301
 - 15.10 深层模型精炼 302
 - 15.11 选择重构目标 302
- 第16章 大型结构 303
 - 16.1 模式：EVOLVING ORDER 306
 - 16.2 模式：SYSTEM METAPHOR 308
 - 16.3 模式：RESPONSIBILITY LAYER 309

16.4 模式 : KNOWLEDGE LEVEL	321
16.5 模式 : PLUGGABLE COMPONENT FRAMEWORK	328
16.6 结构应该有一种什么样的约束	332
16.7 通过重构得到更适当的结构	333
16.7.1 ZUI小化	333
16.7.2 沟通和自律	334
16.7.3 通过重构得到柔性设计	334
16.7.4 通过精炼可以减轻负担	334
第17章 领域驱动设计的综合运用	336
17.1 把大型结构与BOUNDED CONTEXT结合起来使用	336
17.2 将大型结构与精炼结合起来使用	339
17.3 首先评估	339
17.4 由谁制定策略	341
17.4.1 从应用程序开发自动得出的结构	341
17.4.2 以客户为中心的架构团队	341
17.5 制定战略设计决策的6个要点	342
17.5.1 技术框架同样如此	344
17.5.2 注意总体规划	345
结束语	
附录	351
术语表	354
参考文献	357
图片说明	359
索引	360

《领域驱动设计》

精彩短评

- 1、十年前东西，有些不知不觉在用了；边界上下方对微服务有借鉴作用，其它业务领域了解不深
- 2、很多描述的模式已经被应用到框架之中，很多解耦方式已经习以为常，鉴于本书出版的年代，值得钦佩

《领域驱动设计》

精彩书评

1、该书作者显然拥有大量的设计、编码实践。而且看的出，还是敏捷的拥护者。难能可贵的是，该书的翻译质量还是很高的。很多地方直接使用英文原文，而不是搞个蹩脚的中文翻译来打乱你的阅读节奏。只是有部分举例可能因为需要具有业务背景知识才好理解，所以自己感觉没能特别掌握作者要表达的深层含义。

《领域驱动设计》

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:www.tushu000.com