

《实现领域驱动设计》

图书基本信息

书名：《实现领域驱动设计》

13位ISBN编号：9787121224485

10位ISBN编号：7121224488

出版时间：2014-3

作者：Vaughn Vernon

页数：584

译者：滕云

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.tushu111.com

《实现领域驱动设计》

内容概要

领域驱动设计（DDD）是教我们如何做好软件的，同时也是教我们如何更好地使用面向对象技术的。它为我们提供了设计软件的全新视角，同时也给开发者留下了一大难题：如何将领域驱动设计付诸实践？Vaughn Vernon 的这本《实现领域驱动设计》为我们给出了全面的解答。

《实现领域驱动设计》分别从战略和战术层面详尽地讨论了如何实现DDD，其中包含了大量的最佳实践、设计准则和对一些问题的折中性讨论。《实现领域驱动设计》共分为14章，在DDD战略部分，《实现领域驱动设计》向我们讲解了领域、限界上下文、上下文映射图和架构等内容，战术部分包括实体、值对象、领域服务、领域事件、聚合和资源库等内容。一个虚构的案例研究贯穿全书，这对于实例讲解DDD实现来说非常有用。

《实现领域驱动设计》在DDD的思想和实现之间建立起了一座桥梁，架构师和程序员均可阅读，同时也可以作为一本DDD参考书。

《实现领域驱动设计》

作者简介

作者：Vaughn Vernon是一个经验丰富的软件工匠，在软件设计、开发和架构方面拥有超过25年的从业经验。他提倡通过创新来简化软件的设计和实现。从20世纪80年代开始，他便开始使用面向对象语言进行编程；在20世纪90年代早期，他便在领域建模中应用了领域驱动设计，那时他使用的是Smalltalk语言。他在很多业务领域都有从业经验，包括航空、环境、地理、保险、医学和电信等领域。同时，Vaughn在技术上也取得了很大的成功，包括开发可重用的框架和类库等。他在全球范围之内提供软件咨询和演讲，此外，他还在许多国家教授《实现领域驱动设计》的课程。你可以通过www.VaughnVernon.co访问到他的最新研究成果。他的Twitter：[@VaughnVernon](https://twitter.com/VaughnVernon)。

书籍目录

序.....	xix
前言.....	xxi
致谢.....	xxxix
关于作者.....	xxxv
如何使用本书.....	xxxvii
第1章 DDD入门.....	1
我能DDD吗?	2
为什么我们需要DDD.....	5
如何DDD.....	17
使用DDD的业务价值.....	22
1.你获得了一个非常有用的领域模型.....	22
2.你的业务得到了更准确的定义和理解.....	23
3.领域专家可以为软件设计做出贡献.....	23
4.更好的用户体验.....	23
5.清晰的模型边界.....	24
6.更好的企业架构.....	24
7.敏捷、迭代式和持续建模.....	24
8.使用战略和战术新工具.....	24
实施DDD所面临的挑战.....	25
虚构的案例,真实的实践.....	33
本章小结.....	36
第2章 领域、子域和限界上下文.....	37
总览.....	37
工作中的子域和限界上下文.....	38
将关注点放在核心域上.....	42
战略设计为什么重要.....	45
现实世界中领域和子域.....	48
理解限界上下文.....	53
限界上下文不仅仅只包含模型.....	57
限界上下文的大小.....	59
与技术组件保持一致.....	61
示例上下文.....	62
协作上下文.....	63
身份与访问上下文.....	69
敏捷项目管理上下文.....	71
本章小结.....	73
第3章 上下文映射图.....	75
上下文映射图为什么重要.....	75
绘制上下文映射图.....	77
产品和组织关系.....	79
映射3个示例限界上下文.....	82
本章小结.....	97
第4章 架构.....	99
采访一个成功的CIO.....	100
分层.....	104
依赖倒置原则.....	107
六边形架构(端口与适配器)	110

面向服务架构.....	114
REST.....	117
REST作为一种架构风格.....	117
RESTful HTTP服务器的关键方面	118
RESTful HTTP客户端的关键方面	119
REST和DDD.....	120
为什么是REST?	121
命令和查询职责分离——CQRS.....	121
CQRS的各个方面.....	123
处理具有最终一致性的查询模型.....	128
事件驱动架构.....	129
管道和过滤器.....	131
长时处理过程（也叫Saga）.....	134
事件源.....	140
数据网织和基于网格的分布式计算.....	143
数据复制.....	144
事件驱动网织和领域事件.....	145
持续查询.....	145
分布式处理.....	146
本章小结.....	148
第5章 实体.....	149
为什么使用实体.....	149
唯一标识.....	151
用户提供唯一标识.....	152
应用程序生成唯一标识.....	153
持久化机制生成唯一标识.....	156
另一个限界上下文提供唯一标识.....	160
标识生成时间.....	161
委派标识.....	163
标识稳定性.....	165
发现实体及其本质特征.....	167
揭开实体及其本质特征的神秘面纱.....	168
挖掘实体的关键行为.....	172
角色和职责.....	176
创建实体.....	181
验证.....	183
跟踪变化.....	192
本章小结.....	192
第6章 值对象.....	193
值对象的特征.....	194
度量或描述.....	195
不变性.....	195
概念整体.....	196
可替换性.....	199
值对象相等性.....	200
无副作用行为.....	201
最小化集成.....	204
用值对象表示标准类型.....	206
测试值对象.....	210

实现.....	214
持久化值对象.....	219
拒绝由数据建模泄漏带来的不利影响.....	220
ORM与单个值对象.....	221
多个值对象序列化到单个列中.....	224
使用数据库实体保存多个值对象.....	225
使用联合表保存多个值对象.....	229
ORM与枚举状态对象.....	230
本章小结.....	233
第7章 领域服务.....	235
什么是领域服务（首先，什么不是领域服务）.....	237
请确定你是否需要一个领域服务.....	238
建模领域服务.....	241
独立接口有必要吗.....	244
一个计算过程.....	246
转换服务.....	249
为领域服务创建一个迷你层.....	250
测试领域服务.....	250
本章小结.....	253
第8章 领域事件.....	255
何时/为什么使用领域事件.....	255
建模领域事件.....	258
创建具有聚合特征的领域事件.....	263
身份标识.....	264
从领域模型中发布领域事件.....	265
发送方.....	265
订阅方.....	269
向远程限界上下文发布领域事件.....	271
消息设施的一致性.....	271
自治服务和系统.....	272
容许时延.....	273
事件存储.....	274
转发存储事件的架构风格.....	279
以REST资源的方式发布事件通知.....	279
通过消息中间件发布事件通知.....	283
实现.....	284
发布NotificationLog.....	285
发布基于消息的事件通知.....	290
本章小结.....	297
第9章 模块.....	299
通过模块完成设计.....	299
模块的基本命名规范.....	302
领域模型的命名规范.....	302
敏捷项目管理上下文中的模块.....	305
其他层中的模块.....	308
先考虑模块，再是限界上下文.....	309
本章小结.....	310
第10章 聚合.....	311
在Scrum核心领域中使用聚合.....	312

第一次尝试：臃肿的聚合.....	313
第二次尝试：多个聚合.....	314
原则：在一致性边界之内建模真正的不变条件.....	317
原则：设计小聚合.....	319
不要相信每一个用例.....	321
原则：通过唯一标识引用其他聚合.....	322
通过标识引用使多个聚合协同工作.....	324
建模对象导航性.....	325
可伸缩性和分布式.....	326
原则：在边界之外使用最终一致性.....	327
谁的任务？.....	328
打破原则的理由.....	329
理由之一：方便用户界面.....	329
理由之二：缺乏技术机制.....	330
理由之三：全局事务.....	331
理由之四：查询性能.....	331
遵循原则.....	332
通过发现，深入理解.....	332
重新思考设计.....	332
估算聚合成本.....	334
常见用例场景.....	335
内存消耗.....	336
探索另外的设计.....	337
实现最终一致性.....	338
这是Scrum团队成员的任务吗？.....	339
决定的时候到了.....	341
实现.....	341
创建具有唯一标识的根实体.....	342
优先使用值对象.....	343
使用迪米特法则和“告诉而非询问”原则.....	344
乐观并发.....	346
避免依赖注入.....	348
本章小结.....	349
第11章 工厂.....	351
领域模型中的工厂.....	351
聚合根中的工厂方法.....	352
创建CalendarEntry实例.....	353
创建Discussion实例.....	357
领域服务中的工厂.....	358
本章小结.....	361
第12章 资源库.....	363
面向集合资源库.....	364
Hibernate实现.....	369
TopLink实现.....	377
面向持久化资源库.....	379
Coherence实现.....	381
MongoDB实现.....	386
额外的行为.....	391
管理事务.....	393

警告.....	397
类型层级.....	397
资源库 vs 数据访问对象 (DAO)	400
测试资源库.....	401
以内存实现进行测试.....	404
本章小结.....	407
第13章 集成限界上下文.....	409
集成基础知识.....	409
分布式系统之间存在根本性区别.....	411
跨系统边界交换信息.....	411
通过REST资源集成限界上下文.....	417
实现REST资源.....	418
使用防腐层实现REST客户端.....	421
通过消息集成限界上下文.....	428
从Scrum的产品负责人和团队成员处得到持续通知	428
你能处理这样的职责吗?	434
长时处理过程, 以及避免职责.....	439
长时处理过程的状态机和超时跟踪器	450
设计一个更复杂的长时处理过程	460
当消息机制或你的系统不可用时	464
本章小结.....	465
第14章 应用程序.....	467
用户界面.....	469
渲染领域对象.....	470
渲染数据传输对象.....	471
使用调停者发布聚合的内部状态	471
通过领域负载对象渲染聚合实例	472
聚合实例的状态展现.....	473
用例优化资源库查询.....	474
处理不同类型的客户端.....	474
渲染适配器以及处理用户编辑.....	475
应用服务	478
示例应用服务	478
解耦服务输出	485
组合多个限界上下文	487
基础设施	489
企业组件容器	490
本章小结	494
附录A 聚合与事件源: A+ES	495
应用服务内部	496
命令处理器	505
Lambda语法.....	508
并发控制	510
A+ES所带来的结构自由性	513
性能	513
实现事件存储	516
关系型持久化	520
BLOB持久化	522
专注的聚合	523

《实现领域驱动设计》

读模型投射	524
与聚合设计一道使用	527
增强事件	527
工具和模式	529
事件序列器	530
事件不变性	531
值对象	531
协议生成	534
单元测试和需求规范	535
事件源和函数式语言	536
参考文献	539

《实现领域驱动设计》

精彩短评

- 1、太完美了，领域驱动和事件驱动的结合，很好的解决了有界上下文之间数据通讯的问题。
- 2、看了DDD,再看这本IDDD,结合示例代码理解,很多概念更明确了。如果没有WEB开发经验,这本书可能会读起来难理解些。
- 3、读后感：前3章可以略过（免于纸上谈兵），从第4章开始读。最有价值的方面是其中讲到了很多设计与技术实现细节，在实际项目中大都遇到过。关于书中讲到“事件驱动架构（异步-消息机制）”，理论大于实际，忽略了一个比较严重的问题，就是分布式事务管理。就个人目前的项目经验与理解，MQ消息机制更适合作为不同应用间的集成技术。
- 4、DDD讲得还是不错的。找机会试着练练手。
- 5、DDD实战。所在团队实践跟此差别过大吧，感觉不是很好。
- 6、指导意义有，想读下去好难
- 7、增加了一小部分的理解
- 8、看了一些，说得太好了。以前也看了领域驱动设计：软件核心复杂性应对之道，可惜还是不知道怎么下手，看了一些项目示例，可能封装得太深了，没理清头绪，又找了其他面向对象设计的书看了也很朦胧。看了这本感觉豁然开朗。
- 9、一本DDD的实现示例，有很多的代码可以参考，很多抽象的问题都举了具体和现实的例子，很有指导意义
- 10、DDD的绝对力作，教你如何让DDD落地！
- 11、后续再读几遍。这本书一直未能消化彻底
- 12、技术公司的老板们也应该读一读，如果读不懂，建议把公司关了，因为这是迟早的结果。
- 13、对OO的领悟分为三个阶段：
 - 一、开始使用面向对象
 - 二、开始使用设计模式
 - 三、开始使用领域设计
- 14、覆盖面还是可以，提出一套切实可行的设计（比设计模式中的设计更抽象）方法论，但比起基于UML的那套面向对象方法论并没有太多亮点，对于企业和互联网后端开发都适合，在这个云端的时代稍微话题显得有点过时，不过还是值得一看的
- 15、理论性较强，需要在项目中多多锤炼。
- 16、受益匪浅
- 17、内容都是糊在一起的，条理不够清楚
- 18、又把这本书重新读了一遍。一年后经过实践，培训，再读这本书，虽然还有些不明白的地方，但DDD的概念已经逐渐清晰。特别是这次把书中的源代码下下来一起阅读，能完整受到到作者的设计意图，比书中只贴出一个代码片段要好很多。DDD分为战术和战略设计两部分。战术设计比较具体，也好把握。战略设计就相对抽象，对设计和架构的要求就高一些。不管怎么说，实战为王，多参与项目的领域建模，才能真正让DDD在项目中落地。
- 19、真的不怎么样，没有人会按照这个按部就班的实施领域驱动设计吧，也许是我期望太高了。
- 20、虽然还是不太明白领域驱动设计，但是对一些零散的概念已有了一定的认识，希望通过重读、实践、阅读其他书籍能够帮我体会其中的奥秘。
- 21、名为Implementing，感觉离落地太远，也可能自己境界不够，关于架构的讨论比较好看。
- 22、2015-08-24 专注业务架构而不是语言之争。
- 23、有些事情很难实现，会不会太纸上谈兵？
- 24、看完后最大的收获是教会我思考怎么去做抽象。干货非常多的一本书，设计层面和实际编码层面都非常详细。
- 25、阅后感觉对DDD各方面理论理解得更加明确了，但是例子确实不够详细，有些地方翻译的比较晦涩
- 26、粗读了第一遍，了解了很多新概念
- 27、对软件开发中常见的情景进行了很好的抽象，归类。虽然有些地方说得不够透彻，有些地方的说服力不够，还有些地方甚至误导严重（比如利用消息集成限界上下文时，通过事件发生时间来解决顺

《实现领域驱动设计》

序问题的方案，如果消费端部署在多个节点，这个方案就是有问题的，或者很不完备，而这在目前互联网应用中是非常普遍的情况)。但本书对如何应用DDD，如何站在更高的层次进行设计，如何权衡设计，进行了清晰的说明，并提出了切实可行的建议和原则。一直在想老外为什么这么善于抽象、创造新名词，进而建立一套体系/思想/方法

28、浙江省图书馆

29、挺好的一本书，你关注的要点都要，多读几遍

《实现领域驱动设计》

精彩书评

- 1、Eric Evans那本《实现领域驱动设计(DDD)》固然前无古人，但也向我们程序员抛出了一大问题，即如何将DDD付诸实践。十年之后的今天，Vaughn Vernon的这本《实现领域驱动设计》为我们给出了答案。最喜欢的是本书中有关实体(Entity)和资源库(Repository)的讲解，它告诉我们实体所拥有的本质特征，以及实体与值对象(Value Object)的区别。而在读完“第12章 资源库”后，我才发现，自己之前的实现方式都是错的。
- 2、将Evans的DDD和当时现有技术结合落地，是初学者非常棒的DDD入门参考。该书提出的两个聚合引用对方ID的想法比较创新，如果再引入领域事件就更好了，结合 CQRS EventSourcing和 Reactive会更加丰满。

章节试读

1、《实现领域驱动设计》的笔记-限界上下文的大小

为了分配任务而拆分限界上下文是一种错误的上下文建模方式

《实现领域驱动设计》

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.tushu111.com