

Broadview®
www.broadview.com.cn

 Pearson

领域驱动设计 精粹

Domain-Driven Design Distilled

【美】Vaughn Vernon 著
覃宇 笕磊 译

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

领域驱动设计精粹

Domain-Driven
Design Distilled

【美】Vaughn Vernon

重宇 译
图书馆

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

领域驱动设计(DDD)是时下软件设计领域中的热门话题,它通过指导我们构建领域模型,来表达丰富的软件功能需求,并由此实现可以满足用户真正需要的软件。然而在实践中,由于不同的角色对于DDD的核心概念和主要工具的理解不同,常常会造成协作上的不一致。为了帮助和指导面向对象的开发人员、系统分析人员和设计人员更加合理地组织工作,各有侧重、有条不紊地进行复杂系统的开发,并有效地建立丰富而实用的领域模型,本书的作者Vaughn Vernon将自己近年来在领域驱动设计领域的理解进一步提炼,并将本书以精粹的形式呈现给广大的读者。

本书的内容包括:DDD对于广大读者的意义、从战略层面进行设计、从战术层面进行设计,以及相关的辅助工具。

当然,仅仅通过此书的阅读无法深入地掌握领域驱动设计的精髓,无论你是什么经验水平或角色,请阅读本书并在项目中实践DDD。并在这之后,再重读此书,看看你从项目的经历中学到了什么。反复这样的循环,你将会获益匪浅。

Authorized translation from the English language edition, entitled Domain-Driven Design Distilled, ISBN: 9780134434421 by Vaughn Vernon, published by Pearson Education, Inc, Copyright © 2016 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2018

本书简体中文版专有版权由Pearson Education培生教育出版集团授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有Pearson Education培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2016-8519

图书在版编目(CIP)数据

领域驱动设计精粹 / (美)沃恩·弗农(Vaughn Vernon)著;覃宇,笪磊译. —北京:电子工业出版社,2018.9
书名原文:Domain-Driven Design Distilled

ISBN 978-7-121-34852-5

I. ①领… II. ①沃… ②覃… ③笪… III. ①软件设计—研究 IV. ①TP311.1

中国版本图书馆CIP数据核字(2018)第180961号

责任编辑:张春雨

印 刷:三河市华成印务有限公司

装 订:三河市华成印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×980 1/16 印张:10.25 字数:166.3千字

版 次:2018年9月第1版

印 次:2018年9月第1次印刷

定 价:65.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至zlbs@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

本书咨询联系方式:010-51260888-819,faq@phei.com.cn。

译者序

2003年，Eric Evans的《领域驱动设计》出版，第一次总结了这种软件设计和建模方法。这种方法让团队在质疑中发展出对复杂问题的统一认识，再利用战略设计和战术设计的各种手段，如同庖丁解牛般地分解并映射成各种构造块，最后信手拈来地运用各种设计模式将这些构造块一一化解。领域驱动设计在国外的技术社区一直是受到热捧、不断演化的软件设计方法。在Eric的著作面世十年之后，另一位DDD社区的领军人物Vaughn Vernon撰写了《实现领域驱动设计》。在这本著作中，Vaughn用一个连贯完整的实例，将领域驱动设计的所有概念和模式串连在一起，并将这些内容落地到了实例的代码之中。另外，他还在这部著作中总结了这十年来DDD社区涌现的一些新的架构风格和模式，如事件溯源和CQRS、REST风格的架构、事件驱动的架构、六边形架构，等等。

但这十几年间，在国内技术社区，领域驱动设计却像被遗忘在角落的宝藏等待着人们去发掘。当越来越复杂的业务场景开始频繁涌现，当工程实践和基础设施发展成熟，我们重新将视线汇聚在如何达成有效设计、将复杂的业务分而治之，我们发现这种设计方法仿佛早就看透了一切。当宝藏上的灰尘被拂去，领域驱动设计再次发出璀璨夺目的光芒，为我们指明应对软件系统复杂性的前进方向。

重新焕发青春活力的领域驱动设计得到了许多新的团队和架构师的青睐。他们首先就会去阅读这两部略微晦涩的著作，期望能快速地学习和掌握这种方法，但很快就会发现这并不轻松。首先，这两部著作要求读者具备一定的软件开发技术背景。在领域驱动设计的

实践中，业务领域的专家在团队中扮演关键角色，他们往往没有软件开发的技术背景。两位软件巨匠在著作中详细阐述技术概念和实现代码时并没有照顾他们的感受。其次，这两部著作缺少对实际项目建模过程的描写。我们读到的内容多是概念的阐述和与之对应的实例及代码，对于建模实操的过程和工具着墨不多。而这些 Magic Move 却是很多团队实施领域驱动设计时迫切需要指导的关键步骤。最后，两部著作的内容包罗万象，读者容易被繁杂的知识淹没。两部著作中的一些概念和模式（如值对象、实体、工厂和仓储）早已深入人心。而另一部分模式和架构（如事件溯源和 CQRS）则要求架构经验尚浅的读者通过项目实践或扩展阅读才能深入理解。

作为《实现领域驱动设计》一书的作者，Vaughn 也意识到了这些问题，因此编写了这本“精粹版”。他将领域驱动设计的知识进行了提炼，保留了子域、限界上下文、上下文映射、聚合、领域事件这些核心概念，分别用一个章节进行了阐述。在最后一章，作者将他过去在一些团队中实践领域驱动设计时行之有效的具体操作方法（如风靡 DDD 社区的事件风暴工作坊）和工具进行了总结。本书的内容更侧重于高层次的战略设计，关于战术设计的内容偏少，尤其是代码在内容中的比重极低，完全不影响非技术背景的读者阅读。如果你想开始在团队中尝试领域驱动设计，对于团队（包括业务领域的专家）来说，本书的内容可以作为指导手册，让他们快速地进入状态，达到可以参与事件风暴工作坊的要求。我们建议读者们在阅读本书之后亲自组织并实施一次事件风暴工作坊，这是作者推荐的融合视觉、听觉和触觉三种学习方式的“知识获取”实践，是威力无穷的领域建模形式。在开发团队完成建模并最终需要落实到代码时，读者可以将本书作为“武林秘籍”的目录，结合前两部著作和本书参考文献中引用的其他专著一起阅读。

本书中，作者毫不掩饰地表达了对一些架构模式和具体实践的偏好。这些特色鲜明的观点之中，有些符合社区的普遍认知，如事件驱动的响应式架构、单元测试、事件风暴；有些却是对争议性话题的个人理解，如作者对于建模设计的工作量估算的看法。我们要牢记一点，没有“银弹”可以精确地匹配我们的产品和团队，或者完美地解决我们要面对的问题。任何工具和实践都有约束条件。读者们在采用这些工具和实践时，不妨仔细思考作

者运用它们的上下文及其体现出的原则，结合自己的实际情况对工具和实践进行持续改进，避免出现教条主义错误。

我和同事笄磊结对完成了对本书的翻译。我们一人擅长技术，一人则擅长管理，翻译的过程也是我们默契配合、实践“发展通用语言”的“知识获取”过程。我们也将个人对关键内容的理解补充记录在译注中。我们力求翻译内容的准确和译注的质量，但受限于个人经验和知识水平，难免出现偏差甚至错误，还请各位读者斧正。

本书翻译工作于 2017 年末启动，两个月后初稿完成并进入了审校阶段。这期间正值农历戊戌年春节，我们的投入离不开家人们的理解和支持，谢谢她们。我们还要感谢提出宝贵意见的审校者：肖然、刘传湘、王威、朱傲、黄雨清、王林波。他们过去几年都活跃在国内 DDD 社区，也帮助过许多团队运用领域驱动设计方法和事件风暴工作坊来实施架构设计和系统改造。他们过硬的理论知识和丰富的实践经验让本书的翻译增色不少。最后，我们还要感谢专业和严谨的编辑张春雨和刘佳禾，本书也凝聚着你们的心血。

覃宇

2018 年 7 月

序

为什么建模是一件既有趣又回报诸多的事情？从孩童时起，我就喜欢上了构建模型。那时我搭建最多的是汽车和飞机。当时我并不知道乐高玩具的存在。不过，从我的儿子很小的时候起，乐高就一直伴随着他的成长。用这些乐高积木构思和搭建模型是如此令人着迷。构建一个基础模型非常简单，几乎可以无止境地发挥想象力。

你也许也会联想到某些新潮的建筑模型。

模型在我们生活中无处不在。如果你喜欢玩桌游，你正在使用模型。这里的模型可能是房产和业主，或是岛屿和幸存者，抑或是领地和建设活动，等等。同样，视频游戏也藏匿了多个模型。这些模型也许塑造了一个奇幻世界，那里充满了奇特的人物，他们扮演着梦幻般的角色。而扑克牌或者与之相关的游戏则是体现着竞技的模型。模型就是生活的一部分。人们总是在使用模型，虽然往往并不会承认。

为何如此？因为人人皆有各自的学习方式。虽然学习方式有很多种，但讨论最多的三种方式是听觉、视觉和触觉。听觉学习者通过声音和聆听来学习。视觉学习者通过阅读和浏览影像来学习。触觉学习者通过触摸的行为来学习。有趣的是，每个人都会热衷于某种学习方式，以至于有时会在其他学习方式上遇到麻烦。例如，触觉学习者能记住他们所做过的事情，但可能会忘记过程中所听见的内容。建模的过程主要会涉及视觉和触觉的刺激，因此你可能会认为视觉和触觉学习者比听觉学习者更具有优势。然而，事实并非总是如此，尤其当一组建模者在构建的过程中使用有声沟通时。换句话说，模型的构建过程应该适用于大多数人的学习方式。

从建模中学习的能力是人类与生俱来的，为何不利用它去构建已经给生活带来巨大帮助和影响的软件模型呢？事实上，软件模型需要人类去实现，也应该由人类去完成。我认为，人类本应该是优秀的软件模型构建者。

我强烈期望能够帮助你使用最好的建模工具来实现软件。这些工具已被打包成“领域驱动设计”工具箱，或称之为“DDD”工具箱。该工具箱实际上是一套模式，在 Eric Evans 所著的《领域驱动设计：软件核心复杂性应对之道》[DDD]一书中首次提出。我期望将 DDD 带给每一个人。如果必须表达我的观点，我想说的是，让我把 DDD 介绍给大家吧！DDD 也本该如此，它是面向模型设计的人们用于构建卓越软件模型的工具箱。本书中，我会尽可能地简化 DDD 的学习和使用，并将其带给每一位读者。

对于听觉学习者而言，DDD 通过团队的沟通来构建基于通用语言的开发模型，并以此创造学习的契机。对于视觉和触觉学习者来说，在团队进行战略和战术建模时使用 DDD，其过程高度视觉化并非常注重实操。绘制上下文映射图¹并使用事件风暴构建业务流程时尤为如此。因此，我相信 DDD 可以帮助到每一位期待通过模型构建来学习并且希望获得伟大成就的人。

本书所面向的读者²

本书适用于对快速学习 DDD 核心概念和主要工具感兴趣的人。最主要的读者是软件架构师和开发者，他们将在项目中实践 DDD。通常，软件开发者会很快发现 DDD 的美妙之处，并被其强大的工具深深地吸引。尽管如此，本书也可以帮助高管、领域专家、经理人、业务分析师、信息架构师和测试人员理解这一主题。并非只有那些从事信息技术（IT）

1 Context Map, 另有一种常见的译法：上下文地图。此处采用了作者另一本著作《实现领域驱动设计》[IDDD]中的译法：上下文映射图，更好地和上下文映射（Context Mapping）呼应。——译注

2 本书以 DDD 所涵盖的特定概念与工具，期望在产品设计和研发过程所涉及的各类干系人之间搭建一座有效沟通的桥梁，并以此加深人员之间的交互与协作，从而使得最佳的产品设计得以实现。——译注

行业和研发（R&D）行业的从业者才能从书中获益。

如果你是一位顾问，并且正在推荐你的客户使用 DDD，那么请将本书提供给主要负责人，这会帮他们快速理解 DDD。如果你团队中的初级、中级甚至资深开发人员需要尽快在项目上采用 DDD，但对其并不熟悉，请让他们阅读此书。本书至少可以让所有的项目负责人和开发人员熟悉 DDD 词汇表，并了解即将使用的主要工具，这将使他们能在项目推进过程中不断地分享一些有意义的内容。

无论你处于什么样的经验水平或担任什么样的职务，请阅读本书并在项目中实践 DDD。之后再重读此书，看看你从项目中学到了什么，以及将来如何进一步改进。

本书的内容

第 1 章解释了 DDD 能为你和你所在的组织带来什么，并详尽地说明了你将学到什么，以及 DDD 为何如此重要。

第 2 章介绍了 DDD 的战略设计，并教授了 DDD 的重要概念：限界上下文与通用语言。第 3 章解释了子域，以及在新应用建模时如何使用子域应对新应用与现有遗留系统集成的复杂性。第 4 章教授了团队在战略层面上的各种协作方式，以及软件的集成方式，即所谓的上下文映射。

第 5 章将注意力转移到了使用聚合进行战术建模中。领域事件是一个与聚合共同使用的重要而又强大的战术建模工具，它是第 6 章的主题——运用领域事件进行战术设计。

最后，第 7 章会着重介绍一些加速设计和管理项目的工具，它们可以帮助团队建立并保持研发节奏。以上的两个话题很少在其他 DDD 书籍中讨论，但确实是那些决心将 DDD 付诸实践的人所迫切需要的。

行文惯例

请在阅读中注意几点行文惯例。所有讨论的 DDD 工具都会用楷体字表示。例如，你将会看到以楷体印刷的限界上下文和领域事件。另外所有的源代码将会用等宽字体印刷。本书正文章节中出现在方括号内的首字母缩写代表相关的书籍和文献，它们都列在本书末尾的参考文献中。

除此之外，本书重点强调的，也是读者最为喜欢的部分是，通过大量图表和插图进行的视觉学习方式。本书中所有插图都没有任何数字编号，这是因为我不想让太多的数字分散你的注意力。每个案例的图表和插图都会位于对其讨论的正文之前，这也意味着当你阅读本书时，这些视觉图形将首先引发你的思考¹。而当阅读正文时，你也可以反过来参考之前的插图和图表。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- ◎ **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- ◎ **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。



页面入口：<http://www.broadview.com.cn/34852>

¹ 这些视觉图形借助了极其精简的 UML 类图来表达概念，读者即便没有任何 UML 背景知识也很容易理解图形的含义。这些矩形类图可以写成一张张便利贴，读者可以拿着它们站在白板前进行讨论。把它们粘到白板上的任意位置，用马克笔画出圆圈把相关概念圈在一起，或是画出线段连接它们。在这个过程中你们会通过语言大声地交流。这是我们在敏捷实践中经过验证的最高效的协作形式，是视觉、听觉和触觉三种学习方式的完美结合。本书第 7 章将要介绍的事件风暴就是如此。——译注

致谢

这是我和 Addison-Wesley 合作出版的第三本书，同样也是我与编辑 Chris Guzikowski 和 Chris Zahn 的第三次合作。对我而言，这一次和前两次一样令人兴奋。再一次感谢各位选择将我的书籍出版。

与以往一样，本书的顺利完成与出版离不开专家的审阅及反馈。这一次，我将目光转向了那些不需要教授或为 DDD 著书的实干家们，但他们仍然奋斗在项目一线，并帮助其他人使用强大的 DDD 工具箱。我认为，也只有他们这样的实干家才可以确保极度精练的材料以正确的方式精确地表述了必要的内容。这就有点像，如果你希望我做一次 60 分钟的演讲，只需要给我 5 分钟时间去准备；而如果是 5 分钟的演讲，我则需要几个小时来准备。

以下都是给予我莫大帮助的人们（按字母排序）：J r mie Chassaing、Brian Dunlap、Yuji Kiriki、Tom Stockton、Tormod J. Varhaugvik、Daniel Westheide 和 Philip Windley。非常感谢他们！

关于作者

Vaughn Vernon 是一位经验丰富的软件工匠，也是追求简化软件设计和实现的思想领袖。他是畅销书《实现领域驱动设计》和《响应式架构：消息模式 Actor 实现与 Scala、Akka 应用集成》的作者，这些书也同样由 Addison-Wesley 出版发行。他在全球面向数百位开发者教授过 IDDD 课程，并经常在行业会议上发表演讲。他对分布式计算、消息机制，特别是 Actor 模型非常感兴趣。Vaughn 擅长领域驱动设计和使用 Scala、Akka 实现 DDD 方面的咨询。你可以通过 www.VaughnVernon.co 查看他的最新研究成果，或者关注他的 Twitter: @VaughnVernon。

目录

第 1 章 DDD 对我而言	1
DDD 很难掌握吗.....	2
优秀设计、糟糕设计和有效设计	3
战略设计	8
战术设计	9
学习过程与知识提炼	10
让我们开始吧!.....	11
第 2 章 运用限界上下文与通用语言进行战略设计	13
领域专家和业务驱动	20
案例分析	24
战略设计是必要的根基	28
在质疑中统一	32
发展通用语言	38
应用场景	42
如何持续	45
架构	46
本章小结	50

第3章 运用子域进行战略设计	51
什么是子域	52
子域类型	53
应对复杂性	54
本章小结	56
第4章 运用上下文映射进行战略设计	57
映射的种类	60
合作关系	60
共享内核	61
客户—供应商	62
跟随者	63
防腐层	64
开放主机服务	65
已发布语言	65
各行其道	66
大泥球	67
善用上下文映射	69
基于 SOAP 的 RPC	70
RESTful HTTP	72
消息机制	74
上下文映射示例	79
本章小结	83
第5章 运用聚合进行战术设计	85
为什么使用它	86
聚合的经验法则	91
规则一：在聚合边界内保护业务规则不变性	92

规则二：聚合要设计得小巧	93
规则三：只能通过标识符引用其他聚合	95
规则四：利用最终一致性更新其他聚合	96
建立聚合模型	99
慎重选择抽象级别	104
大小适中的聚合	106
可测试的单元	108
本章小结	108
第 6 章 运用领域事件进行战术设计	111
设计、实现并运用领域事件	113
事件溯源	119
本章小结	121
第 7 章 加速和管理工具	123
事件风暴	124
其他工具	134
在敏捷项目中管理 DDD	135
运用 SWOT 分析法	137
建模 Spike 和建模债务	139
任务识别与工作量估算	140
限制建模时间	143
如何实施	144
和领域专家打交道	145
本章小结	147
参考文献	148

第 1 章

DDD对我而言

如果你希望打磨软件匠艺并提高项目的成功率，如果你迫切期望创造软件来帮助企业把业务竞争力提升到新高度，如果你期望实现出来的软件既能正确地对业务需求建模又可以采用最先进的软件架构进行扩展，那么就学习并掌握领域驱动设计（DDD）吧。它可以帮你实现这些目标并带来更多收获。

DDD 同时提供了战略和战术上的建模工具，来帮助你设计和实现高价值的软件。你的组织无法在所有方面都出类拔萃，所以最好谨慎地识别其真正的优势所在。DDD 的战略设计工具可以帮助你和你团队做出最有竞争力的软件设计选择和业务整合决策。你的组织将从这些明确反映其核心业务竞争力的软件模型中获得最大的收益。DDD 的战术实施工具可以帮助你和你团队设计出实用的软件，它能对业务独一无二的运作方式进行精准地建模。你的组织可以有更多的选择，把解决方案部署到各种不同的基础设施中，不管这些基础设施是位于企业内部还是在公有云上，你的组织都会受益匪浅。借助 DDD，你的团队可以在当今竞争激烈的商业环境中持续地交付最有效的软件设计与实现。

本书提炼了 DDD 的精华，内容包括了浓缩的战略和战术建模工具。我能理解软件开发过程中出现的种种独特诉求，也对你在快节奏的行业中磨炼技艺时面临的种种挑战感

同身受。你无法花费数月时间学习一门像 DDD 这样的课程，然而你却盼望着尽早在日常工作中运用这一技艺。

我不仅是畅销书《实现领域驱动设计》[IDDD]的作者，也曾创办并教授为期三天的 IDDD 工作坊课程。如今，本书将以浓缩的形式为大家呈现 DDD 的精华。我将致力于让所有的开发团队都能用上 DDD。当然，你也应当在其中。



DDD 很难掌握吗

你或许听说 DDD 是一种难以理解的软件开发方法。但事实并非如此。事实上 DDD 是在复杂的软件工程项目中引入的一系列先进的技术。在没有专家指导的情况下，单靠自己