领域驱动设计 软件核心复杂性应对之道 修订版

作者：[美] 埃里克 埃文斯（Eric Evans）

目录

其他

DOMAIN-DRIVEN DESIGN TACKLING COMPLEXITY IN THE HEART OF SOFTWARE

领域驱动设计　软件核心复杂性应对之道

[美]Eric Evans◎著

赵俐　盛海艳　刘霞　等◎译

任发科◎审校

人民邮电出版社

北京

图书在版编目（CIP）数据

领域驱动设计：软件核心复杂性应对之道/（美）埃文斯（Evans,E.）著；赵俐等译.--2版（修订本）.--北京：人民邮电出版社，2016.6

书名原文：Domain-Driven Design:Tackling Complexity in the Heart of Software

ISBN 978-7-115-37675-6

I.①领… II.①埃…②赵… III.①软件设计 IV.①TP311.5

中国版本图书馆CIP数据核字（2016）第069725号

内容提要

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最佳实践综合到一起，融入了作者的见解和经验，展现了一些可扩展的设计最佳实践、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。

本书适合各层次的面向对象软件开发人员、系统分析员阅读。

◆著 [美]Eric Evans

译 赵俐 盛海艳 刘霞 等

审校 任发科

责任编辑 杨海玲

责任印制 焦志炜

◆人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 http://www.ptpress.com.cn

三河市海波印务有限公司印刷

◆开本：800×1000 1/16

印张：24.25

字数：515千字 2016年6月第2版

印数：1-3000册 2016年6月河北第1次印刷

著作权合同登记号 图字：01-2010-0695号

定价：69.00

读者服务热线：(010)81055410 印装质量热线：(010)81055316

反盗版热线：(010)81055315

版权声明

译者序

我最早听说Eric Evans的《领域驱动设计》是在2007年，那时我所在的项目组出于知识储备的考虑购进了一批软件设计书和相关资料。其中一篇英文的短篇技术文档与我们当时的项目非常相关，于是我们就仔细研读了一番。这篇仅有几万字的文档多次提到了Eric Evans的《领域驱动设计》，并引用了他的很多精辟观点。由于当时领域驱动设计远远没有现在这样普及，因此这些观点使我耳目一新，也给我留下了深刻的印象。随后我又经常在一些文献中看到Eric Evans的名字，更多地了解了他的领域驱动设计思想，没想到时隔几年后竟然有机会把这位大师的作品翻译出来奉献给各位读者，也算是机缘巧合了。

相信大家对这本书都不陌生，它已经成为软件设计书中的经典。在网上搜索一下，读者对它好评如潮，我再多说一句赞美的话都是多余的。而我能想到的也唯有“经典”二字，它堪称经典中的经典。

我们对“领域”这个概念都很熟悉，但有多少人真正重视过它呢？软件开发人员几乎总是专注于技术，把技术作为自己能力的展示和成功的度量。而直到Eric Evans出版了他的这部巨著之后，人们才真正开始关注领域，关注核心领域，关注领域驱动的设计，关注模型驱动的开发。相信在读完本书后，你会对软件设计有全新的认识。

我曾经和一些好友探讨过以下一些问题。项目怎样开发才能确保成功？什么样的软件才能为用户提供真正的价值？什么样的团队才算是优秀的团队？现在，在仔细研读完本书后，这些问题都找到了答案。

本书广泛适用于各种领域的软件开发项目。在每个项目的生命周期中，都会有一些重大关头或转折点。如何制定决策，如何把握项目的方向，如何处理和面对各种机会和挑战，将对项目产生决定性的影响。让我们一起跟随大师的脚步，分享他通过大量项目获得的真知灼见和开发心得吧。

最后，衷心感谢人民邮电出版社各位编辑在翻译工作中给予的帮助和宝贵意见，感谢热心读者魏海枫，他在百忙之中抽出时间对本书译稿做了修订工作，发现并修正了很多问题。由于译者水平有限，在翻译过程中难免还会留有一些错误，恳请读者批评指正。

序

有很多因素会使软件开发复杂化，但最根本的原因是问题领域本身错综复杂。如果你要为一家人员复杂的企业提高自动化程度，那么你开发的软件将无法回避这种复杂性，你所能做的只有控制这种复杂性。

控制复杂性的关键是有一个好的领域模型，这个模型不应该仅仅停留在领域的表面，而是要透过表象抓住领域的实质结构，从而为软件开发人员提供他们所需的支持。好的领域模型价值连城，但要想开发出好的模型也并非易事。精通此道的人并不多，而且这方面的知识也很难传授。

Eric Evans就是为数不多的能够创建出优秀领域模型的人。我是在与他合作时发现他的这种才能的——发现一个客户竟然比我技术更精湛，这种感觉有些奇妙。我们的合作虽然短暂，但却充满乐趣。从那之后我们一直保持联系，我也有幸见证了本书整个“孕育”过程。

本书绝对值得期待。

本书终于实现了一个宏伟抱负，即描述并建立了领域建模艺术的词汇库。它提供了一个参考框架，人们可以用它来解释相关活动，并用它来传授这门难学的技艺。本书在写作过程中，也带给我很多新想法，如果哪位概念建模方面的老手没有从阅读本书中获得大量的新思想，那我反而该惊诧莫名了。

Eric还对我们多年以来学过的知识进行了归纳总结。首先，在领域建模过程中不应将概念与实现割裂开来。高效的领域建模人员不仅应该能够在白板上与会计师进行讨论，而且还应该能与程序员一道编写Java代码。之所以要具备这些能力，一部分原因是如果不考虑实现问题就无法构建出有用的概念模型。但概念与实现密不可分的最主要原因在于，领域模型的最大价值是它提供了一种通用语言，这种语言是将领域专家和技术人员联系在一起的纽带。

我们将从本书中学到的另一个经验是领域模型并不是按照“先建模，后实现”这个次序来工作的。像很多人一样，我也反对“先设计，再构建”这种固定的思维模式。Eric的经验告诉我们，真正强大的领域模型是随着时间演进的，即使是最有经验的建模人员也往往发现他们是在系统的初始版本完成之后才有了最好的想法。

我衷心希望本书成为一本有影响力的著作，并希望本书能够将如何利用领域模型这一宝贵工具的知识传授给更多的人，从而为这个高深莫测的领域梳理出一个结构，并使它更有内聚力。领域模型对软件开发的控制有着巨大影响，不管软件开发是用什么语言或环境实现的。

最后，也是很重要的一点，我最敬佩Eric的一点是他敢于在本书中谈论自己的一些失败经历。很多作者都喜欢摆出一副无所不能的架势，有时着实让人不屑。但Eric清楚地表明他像我们大多数人一样，既品尝过成功的美酒，也体验过失败的沮丧。重要的是他能够从成功和失败中学习，而对我们来说更重要的是他能够将所有经验传授给我们。

Martin Fowler

2003年4月