

<<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：9787115238870

10位ISBN编号：7115238871

出版时间：2010-11

出版时间：人民邮电出版社

作者：埃文斯

页数：369

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，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<领域驱动设计>>

前言

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 Evsns就是为数不多的一位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，人们可以用来解释相关活动，并将这门难学的技艺传授给他人。

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。

<<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。

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。

<<领域驱动设计>>

作者简介

作者：（美国）埃文斯（Eric Evans）译者：赵俐 盛海艳 刘霞 等Eric Evans，世界著名软件建模专家，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。他在全球各地宣讲领域驱动设计的思想，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。

<<领域驱动设计>>

书籍目录

| | | | |
|-------------------------------------|----------------------------|----------------------|------------------------------------|
| 第一部分 让领域模型发挥作用 | 第1章 消化知识 | 1.1 有效建模的要素 | 1.2 知识消化 |
| 1.3 持续学习 | 1.4 知识丰富的设计 | 1.5 深层模型 | 第2章 语言的交流和使用 |
| 模式：UBIQUITOUS LANGUAGE | 2.2 “大声地”建模 | 2.3 一个团队，一种语言 | 2.4 |
| 文档和图 | 2.4.1 书面设计文档 | 2.4.2 完全依赖可执行代码的情况 | 2.5 解释性模型 |
| 第3章 绑定模型和实现 | 3.1 模式：MODEL-DRIVEN DESIGN | 3.2 建模范式和工具支持 | 3.3 揭示主旨：为什么模型对用户至关重要 |
| 3.4 模式：HANDS-ON MODELER | 第二部分 模型驱动设计的构造块 | 第4章 分离领域 | 4.1 模式：LAYERED ARCHITECTURE |
| 4.1.1 将各层关联起来 | 4.1.2 架构框架 | 4.2 模型属于领域层 | 4.3 模式：THE SMART UI “ANTI-PATTERN” |
| 4.4 其他分离方式 | 第5章 软件中所表示的模型 | 5.1 关联 | 5.2 模式：ENTITY(又称为REFERENCE OBJECT) |
| 5.2.1 ENTITY建模 | 5.2.2 | 5.3 模式：VALUE OBJECT | 5.3.1 设计VALUE OBJECT |
| 5.3.2 | 5.4 模式：SERVICE | 5.4.1 SERVICE与孤立的领域层 | 5.4.2 粒度 |
| 5.4.3 对SERVICE的访问 | 5.5 模式：MODULE(也称为PACKAGE) | 5.5.1 敏捷的MODULE | 5.5.2 基础设施驱动的打包存在的隐患 |
| 5.5.3 在混合范式中坚持使用MODEL-DRIVEN DESIGN | 第6章 领域对象的生命周期 | 6.1 模式：AGGREGATE | ... |
| 第7章 使用语言：一个扩展的示例 | 第三部分 通过重构来加深理解 | 第8章 突破 | 第9章 将隐式概念转变为显式概念 |
| 第10章 柔性设计 | 第11章 分析模式的应用 | 第12章 将设计模式应用于模型 | 第13章 通过重构得到更深层的理解 |
| 第四部分 战略设计 | 第14章 保持模型的完整性 | 第15章 精炼 | 第16章 大比例结构 |
| 第17章 领域驱动设计的综合运用 | 结束语附录 | 术语表 | 参考文献 |
| 图片说明 | 索引 | | |

<<领域驱动设计>>

章节摘录

插图：随着系统的增长，它会变得越来越复杂，当我们无法通过分析对象来理解系统的时候，就需要掌握一些操纵和理解大模型的技术了。

本书的这一部分将介绍一些原则。

遵循这些原则，就可以对一些十分复杂的领域进行建模。

大部分这样的决策都需要由团队来制定，甚至需要多个团队共同协商制定。

这些决策往往是把设计和策略综合到一起的结果。

最卓越的企业系统的目标是实现一个把所有业务都包括进来的、紧密集成的系统。

然而在几乎所有这种规模的组织中，整体业务模型太大也太复杂了，因此难以管理，甚至很难把它作为一个整体来理解。

我们必须在概念和实现上把系统分解为较小的部分。

问题是如何在不损害集成利益的前提下完成这种模块化的过程，从而使系统的不同部分能够进行互操作，以便使各种业务操作互相协调。

如果设计一个把所有概念都涵盖进来的单一领域模型，它将会非常笨拙，而且将会出现大量难以察觉的重复和矛盾。

而如果用一些特定接口把一组小的、各自不同的子系统集成到一起，又会影响解决企业级问题的能力，并且在每个集成点上都有可能出现不一致问题。

通过采用一种系统的、不断演变的设计策略，就可以避免这两种极端问题。

<<领域驱动设计>>

媒体关注与评论

“这本书应该出现在每个软件开发人员的书架上。

”——Kent Beck 软件开发方法学泰斗，极限编程的创始人 “Eric的这本书太棒太神奇了，他准确地告诉你如何让软件设计满足你理想中的模型需求。

.....本书读起来趣味无穷，Eric有许多有趣的故事，而且描述起来很有一套。

它出版后将作为软件开发人员必读的经典之作。

”——Ralph Johnson 《设计模式》的作者 “如果你认为自己在面向对象编程中的投资没有收到回报，那么读了本书你就会知道自己漏掉了什么。

”——Ward Cunningham 设计模式和敏捷软件方法的先驱 “Eric Evarts力证作为开发核心的领域模型的重要性，他搭建了一个稳固的框架并提供了一套实现技术和技巧。

这里沉淀下来的是亘古不变的智慧，在应时的方法论都沦为明日黄花后，它依然光华璀璨。

”——Dave Collins Designing Object-Oriented User Interfaces的作者 “Eric完全从实战者的角度着笔，描述了无处不用的语言、与用户共享模型的好处、对象生命周期的管理、深度重构的过程和结果，这是对我们这个领域的巨大贡献。

”——Luke Hohmann Beyond Software Architecture的作者

<<领域驱动设计>>

编辑推荐

《领域驱动设计:软件核心复杂性应对之道》:众多世界级软件大师鼎力推荐凝聚领域建模专家数十年的实战经验深度剖析构建高质量复杂系统的核心技术《领域驱动设计》与《企业应用架构模式》两大名著精髓的实战演练领域模型使开发人员可以表达丰富的软件功能需求,由此实现的软件可以满足用户真正的需要,因此被公认为是软件设计的关键所在,其重要性显而易见。

但讲述如何将领域模型用于软件开发过程的优秀实用资料却不多见。

《领域驱动设计:软件核心复杂性应对之道》正是这一领域最著名的作品,受到众多业界大师的赞美和推介,广受读者好评。

要通过创建领域模型来加速复杂的软件开发,就需要利用大量最佳实践和标准模式在开发团队中形成统一的交流语言;不仅重构代码,而且要重构代码底层的模型;同时采取反复迭代的敏捷开发方法,深入理解领域特点,促进领域专家与程序员的良好沟通。

针对这些内容,《领域驱动设计:软件核心复杂性应对之道》结合真实项目,系统地介绍了领域驱动开发的目标、意义和方法,充分讨论了复杂系统的建模与设计问题。

《领域驱动设计:软件核心复杂性应对之道》将指导面向对象开发人员、系统分析人员和设计人员合理地组织工作,各有侧重、彼此协作,有条不紊地进行复杂系统的开发,帮助他们建立丰富而实用的领域模型,并由此创建长期适用的优质软件。

Eric Evans强调要聚焦于软件的核心领域,以它来驱动开发。

软件能够在市场上卖出去,是因为它封装了别的软件所没有的一些核心领域知识、这就是核心竞争力,是利润所在的地方,也是最值得下工夫的地方,再难也不能逃避。

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>