## 「战术」领域驱动设计的不确定性

### 课程背景

大家好，我是张逸，去年在 GitChat 平台上上线了一门[《领域驱动战略设计实践》]达人课，销量已过 4000 份，同时建了两个读者群，也邀请了十几位领域驱动设计方面的专家加入到了读者群，共同探讨和交流领域驱动设计的相关知识。

至今，两个群依然很活跃，每天都有许多问题抛出，同时又有许多问题得到了解答，还有更多的问题悬而未决，因为每个人都有自己心目中的“哈姆雷特”，谁也无法说服对方，谁也无法给出一个让所有人都认同的标准答案。这恰恰是领域驱动设计最棘手的一部分，当然，也是最让人神往的一部分—— **唯有不确定，方才值得去探索** 。

在探讨领域驱动战术设计的一些问题时，总会有人纠结：这个领域对象应该定义成实体，还是值对象？领域服务和应用服务的区别是什么？聚合的边界该怎么划分？于是，各种设计问题纷至沓来，问题越辩越糊涂，到了最后，已经脱离了最初探讨问题的场景，变成了“空对空导弹”一阵乱发射，最后蓦然回首，才发现目标已然消失了。

这是不合理的。在软件开发领域，没有什么一劳永逸的实现，也没有什么放之四海而皆准的标准，必须结合具体的业务场景做出合理的决策，无论建模和设计再怎么完美，也需要通过落地的检验才知道好还是坏。任何脱离具体业务场景的问题分析，都是空谈；任何不落地的完美方案，都是浮夸。 **领域驱动设计没有标准，有的只是持续不断的不确定性。**

正所谓“以不变应万变”，我们要从实证主义的角度看待领域驱动设计，窃以为，只需守住三项基本原则即可：

* 必须通过领域建模来驱动设计
* 领域专家或业务分析师必须参与到建模活动中
* 设计必须遵循面向对象分析和设计的思想与原则

只要做到这三点，领域驱动战术设计就不会做得太差，剩下的不足，就需要靠经验来填补了。

[点击了解《领域驱动设计实践（合订版）》](https://gitbook.cn/m/mazi/comp/column?columnId=5cdab7fb34b6ed1398fd8de7&utm_source=zyhd001)

### 课程框架

本课程是我计划撰写的领域驱动设计实践系列的 **第二部分** 课程，要解决的正是前面所提及的战术层面的设计问题。单以 **战术设计阶段** 来看，我个人认为 Eric Evans 做出的贡献并不多。在《领域驱动设计》一书中，他讲到了模型驱动设计与领域建模，却没有深入阐述该如何正确地进行领域建模；他引入的资源库模式和工厂模式，不过是面向对象设计原则的体现；至于模型的演化与重构带来的突破，其实更多是经验之谈，缺乏切实可行的方法。整体而言，Eric Evans 在战术设计要素方面，最为重要的洞见在于：

* 强调了领域建模的重要性，并与面向对象分析和设计的原则结合起来
* 实体与值对象的区分，有利于我们明白模型对象的真相，并能够更好地避免贫血模型
* 聚合提出了有别于模块粒度的边界，有效地保证了业务规则的不变性和数据的一致性

不可否认，若要做到优良的领域驱动设计，建模和设计的经验是必不可少的，这需要多年的项目实战打磨方可萃取而成，但如果在开始之初，能有一些更为具体的方法作为指引，或许可以让掌握技能的周期大幅度缩短。此外，我还清楚地看到：许多领域驱动设计的门外汉，之所以迟迟不得其门而入，是因为他（她）们连最为基本的面向对象分析和设计的能力都不具备，因此，无法理解领域驱动的战术设计要素也就不足为奇了。关键在于， **许多设计问题因为其不确定性，根本没有标准答案，没有任何人能给你指出明确的设计方法和设计思路** 。这时，就必须要吃透面向对象分析和设计的思想与原则，用它们来指导我们的设计，而不是死板的遵循领域驱动设计的模式。

针对一些设计能力不足的开发团队，若希望采用领域驱动设计来改进设计和编码质量，往往会适得其反，做出来的是一锅“夹生饭”。从理想角度讲，决定是否采用领域驱动设计，不在于团队成员的能力高低，而在于业务的复杂度。然而，我们又不得不面对现实，如果团队成员的设计能力差了，是做不好领域驱动设计的。因此，我在本课程中，一方面分享了我的设计体验和方法，以帮助团队成员的成长，另一方面也给出了一个操作性强的设计过程，可以让基础相对薄弱的开发人员能够依样画葫芦，做出还算不错的设计与实现。

这些考虑帮助我确定了本课程的基本思路，即以能学习和模仿的战术设计方法来弥补经验之不足，以设计思想和设计原则作为指导来解决争议之问题，以能够落地的解决方案来体现领域驱动设计之价值。

本课程分为六部分，共 64 篇（含访谈录、开篇词）。

* 第一部分：软件系统中的模型（第 1-1 课 ~ 第 1-15 课）

全面讲解和对比软件系统的数据模型、服务模型和领域模型，以这些模型作为不同的设计驱动力，讲解不同的模型驱动设计之过程与利弊，从而得出领域模型驱动设计的优势以及它适用的业务场景。

* 第二部分：领域分析模型（第 2-1 课 ~ 第 2-7 课）

建立领域分析模型是领域模型驱动设计的起点和基础。领域分析过程是领域专家与开发团队合作最为紧密、沟通最为频繁的阶段，是领域驱动设计成败的关键。我将深入介绍名词动词法、分析模式、四色建模和事件风暴等重要的分析建模方法，在发现显式和隐式领域概念的基础上，建立高质量的领域分析模型。

* 第三部分：领域设计模型（第 3-1 课 ~ 第 3-17 课）

实体、值对象、领域服务、领域事件、资源库、工厂和聚合是组成领域驱动战术设计的核心内容，也是衡量领域驱动设计质量的分水岭。只有正确地理解了这些设计要素，才能正确地完成领域驱动战术设计。这其中扮演关键角色的其实是面向对象分析与设计。我将围绕着职责驱动设计讲解角色、职责与协作三者之间的关系，通过分辨职责来寻找合理的对象，并结合 DCI 模式与主流设计模式，以时序图作为主要的设计驱动力获得高质量的设计方案。

* 第四部分：领域实现模型（第 4-1 课 ~ 第 4-6 课）

领域实现模型帮助我们将领域设计模型落地，毕竟，只有交付可工作的软件才是软件开发的终极目标。除了要应对纷繁复杂的业务逻辑，我们还需要考虑如何与外部资源集成，实现数据持久化与消息通信等基础设施内容。在落地过程中，我们需要时刻维护业务复杂度与技术复杂度的边界，降低彼此的影响，同时还需要在编码层次提高代码的内建质量，包括代码的可读性、可重用性、可扩展性和可测试性。

* 第五部分：融合：战略设计和战术设计（第 5-1 课 ~ 第 5-9 课）

领域驱动设计虽然分为战略设计阶段和战术设计阶段，但这两个阶段并非完全割裂的井水不犯河水的独立过程。我在[《领域驱动战略设计实践》]课程中介绍领域驱动设计过程时，就提到了这两个阶段的相辅相成与迭代的螺旋上升演进过程。我们必须将战略设计和战术设计融合起来，把分层架构、限界上下文、上下文映射与战术设计的诸要素融汇贯通，才能获得最佳的设计质量，并成为指导我们进行软件架构和设计的全过程。

* 第六部分：EAS 系统的战术设计实践（第 6-1 课 ~ 第 6-8 课）

继续沿用战略设计实践中使用的全真案例——EAS 系统，采用领域模型驱动设计的过程对系统进行分析建模、设计建模和实现建模，并最终结合战略设计的方案，形成完整的解决方案和代码实现。

综上，本课程的内容并未完全遵照 Eric Evans 的《领域驱动设计》，不同的部分固然是我的一孔之见，未必正确，也未必遵守 Eric Evans 的设计思想，但我仍然不揣冒昧地进行了分享，不是因为我的无知者无畏，而是我认为针对具有不确定性的领域驱动设计，必须要容得下异见者，方能取得发展和突破。

### 为什么要学习领域驱动设计

如果你已经能设计出美丽优良的软件架构，如果你只希望脚踏实地做一名高效编码的程序员，如果你是一位注重用户体验的前端设计人员，如果你负责的软件系统并不复杂，那么，你确实不需要学习领域驱动设计！

领域驱动设计当然并非“银弹”，自然也不是解决所有疑难杂症的“灵丹妙药”，请事先降低对领域驱动设计的不合现实的期望。我以中肯地态度总结了领域驱动设计可能会给你带来的收获：

* 领域驱动设计是一套完整而系统的设计方法，它能给你从战略设计到战术设计的规范过程，使得你的设计思路能够更加清晰，设计过程更加规范；
* 领域驱动设计尤其善于处理与领域相关的高复杂度业务的产品研发，通过它可以为你的产品建立一个核心而稳定的领域模型内核，有利于领域知识的传递与传承；
* 领域驱动设计强调团队与领域专家的合作，能够帮助团队建立一个沟通良好的团队组织，构建一致的架构体系；
* 领域驱动设计强调对架构与模型的精心打磨，尤其善于处理系统架构的演进设计；
* 领域驱动设计的思想、原则与模式有助于提高团队成员的面向对象设计能力与架构设计能力；
* 领域驱动设计与微服务架构天生匹配，无论是在新项目中设计微服务架构，还是将系统从单体架构演进到微服务设计，都可以遵循领域驱动设计的架构原则。

### 课程寄语

没有谁能够做到领域驱动设计的一蹴而就，一门课程也不可能穷尽领域驱动设计的方方面面。从知识的学习到知识的掌握，进而达到能力的提升，需要一个漫长的过程。所谓“理论联系实际”虽然是一句大家耳熟能详的老话，但其中蕴含了颠扑不破的真理。我在进行领域驱动设计培训时，总会有学员希望我能给出数学公式般的设计准则或规范，似乎软件设计就像拼积木一般，只要遵照图示中给出的拼搭过程，不经思考就能拼出期待的模型。——这是不切实际的幻想。

要掌握领域驱动设计，就不要被它给出的概念所迷惑，而要去思索这些概念背后蕴含的原理，多问一些为什么。同时，要学会运用设计原则去解决问题，而非所谓的“设计规范”。我强烈建议读者诸君要 **学会对设计的本质思考，不要只限于对设计概念的掌握，而要追求对设计原则与方法的融汇贯通** 。只有如此，才能针对不同的业务场景灵活地运用领域驱动设计，而非像一个牵线木偶般遵照着僵硬的过程进行死板地设计。

### 分享交流

我们为本课程付费读者创建了微信交流群，以方便更有针对性地讨论课程相关问题。入群方式请到第 6-1 课末尾添加 小编的微信号，并注明「DDD」。

阅读文章过程中有任何疑问随时可以跟其他小伙伴讨论，或者直接向作者提问（作者看到后抽空回复）。 **你的分享不仅帮助他人，更会提升自己。**

[点击了解《领域驱动设计实践（合订版）》](https://gitbook.cn/m/mazi/comp/column?columnId=5cdab7fb34b6ed1398fd8de7&utm_source=zyhd001)

下一章

还没有评论



评论

## 更多资源下载交流请加微信：Morstrong,加入永久会员,网盘更新更快捷！

# 本资源由微信公众号：光明顶一号，提供支持