## 「战略」访谈录 | 聊聊领域驱动设计（文字版）

相信很多朋友对领域驱动设计会有这样或那样的困惑，比如领域驱动设计是什么？它在工作中有什么作用？为什么国内关于这方面的书籍少之又少？…… 为了解决这些疑惑，有幸邀请到专家张逸老师来聊聊领域驱动设计，下面是 GitChat 独家采访记录。

GitChat：领域驱动设计（Domain Driven Design，DDD）自诞生以来已有十几年时间，这门本已步入老年的方法学却因为 **微服务** 的兴起而焕发了第二春。您说过这可能要归功于 DDD 的“坚硬生长”，但不可否认微服务确实也是一个重要因素，能否请您解释一下 **领域驱动设计** 和 **微服务** 这种深层次的匹配关系？

张逸：领域驱动设计是由 Eric Evans 在一本《领域驱动设计》书中提出的，它是针对复杂系统设计的一套软件工程方法；而微服务是一种架构风格，一个大型复杂软件应用是由一个或多个微服务组成的，系统中的各个微服务可被独立部署，各个微服务之间是松耦合的，每个微服务仅关注于完成一件任务并很好地完成该任务。

两者之间更深入的关系，在我写的课程中已有详细讲解。主要体现在领域驱动设计中限界上下文与微服务之间的映射关系。假如限界上下文之间需要跨进程通信，并形成一种零共享架构，则每个限界上下文就成为了一个微服务。在微服务架构大行其道的当今，我们面临的一个棘手问题是：如何识别和设计微服务？领域驱动的战略设计恰好可以在一定程度上解决此问题。

GitChat：如果说轻量化处理、自动部署，以及容器技术的发展使得微服务的兴起成为必然，那么是否可以说领域驱动设计今日的再续辉煌也是一种必然（或者说 DDD 在其诞生之时过于超前）？您能否预测一下 DDD 未来可能会和什么样的新理念相结合？

张逸：好像领域驱动设计就从未真正“辉煌”过，所以谈不上再续辉煌，但确实是因为微服务引起了社区对它的重燃热情。推行领域驱动设计确乎有许多阻力，一方面要做到纯粹的领域驱动设计，许多团队成员的技能达不到；另一方面，似乎领域驱动设计带来的价值不经过时间的推移无法彰显其价值，这就缺乏足够的说服力让一家公司不遗余力地去推广领域驱动设计。微服务似乎给了我们一个推动领域驱动设计的理由！因为软件系统的微服务化已经成为了一种潮流，领域驱动设计又能够为微服务化保驾护航，还有什么理由不推行呢？

我个人认为，未来 DDD 的发展可能会出现以下趋势：

* 以函数式编程思想为基础的领域建模理念与事件驱动架构和响应式编程的结合，可能在低延迟高并发的项目中发挥作用。这种领域驱动设计思想已经比较成熟，但目前还没有看到太多成功的运用。
* 以 DDD 设计方法为基础的框架的出现，让微服务设计与领域建模变得更加容易，降低领域驱动设计的门槛。

GitChat：能否尽可能地详细（或举例）说明您在阅读并审校《实现领域驱动设计》一书时所认识到的领域驱动设计的本质—— **一个开放的设计方法体系** ——是什么？

张逸：在《实现领域驱动设计》一书中，Vernon 不仅对整个领域驱动设计过程作了一番有益的梳理，还结合社区发展在书中引入了六边形架构和领域事件等概念，这为当时的我打开了一扇全新的窗户——原来领域驱动设计并不是一套死板的方法，而是一种设计思想、一种开放的设计方法体系，只要有利于领域驱动设计的实践，都可以引入其中。于是，在我的书中我才敢于大胆地引入用例、敏捷实践、整洁架构，以期为领域驱动设计提供补充。

Eric Evans 的《领域驱动设计》是以面向对象设计作为模型驱动设计的基础，但时下被频繁运用的函数式编程思想也给模型驱动设计带来了另一种视角。从开放的设计方法体系的角度讲，我们完全可以把更多的编程范式引入到领域驱动设计中。因为有了更多的选择，针对不同的业务场景就可以选择更适合的 DDD 实践，而不仅仅限于 Eric Evans 最初提出的范畴。

[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)。

GitChat：团队内外成员之间的协作与沟通一直以来都是个难题，也是大家经常喜欢调侃的话题之一，能否举例说明一下领域驱动设计是如何解决这一问题的？

张逸：我觉得这个问题问反了。领域驱动设计解决不了这个问题，它只是重视这个问题；相反，我们应该说只有解决了团队内外成员之间的协作与沟通，才能更好地进行领域驱动设计。为此，我尝试用一些敏捷实践来解决这种协作问题。

GitChat：您在学习和实践领域驱动设计的过程中是否有哪些（有趣的）故事可以和读者们分享？

张逸：我在 ThoughtWorks 的时候，公司邀请《实现领域驱动设计》作者 Vaughn Vernon 到北京 Office 给我们做了一次 DDD 培训。借着这次亲炙大师教诲的机会，我向他请教了一个一直缠绕在我心中困惑不解的问题：“如何正确地识别限界上下文？”结果他思考了一会儿，严肃地回答了我：“By experience！” 我唯有无言以对。

GitChat：有很多读者对您即将在课程中给出全真案例“EAS 系统”很感兴趣，能否简单介绍一下这个案例以及它在实际应用中的意义？

张逸：EAS 系统是我之前做过的一个真实项目，之所以选择这个项目来作为这个课程的全真案例，原因如下：

* 学习 DDD 必须理论联系实际。虽然在我写的课程内容中已经结合理论讲解提供了较多的实际案例，但这些零散的案例无法给读者提供一个整体的印象。
* EAS 系统的业务知识门槛相对较低，不至于因为不熟悉领域知识而影响对 DDD 的学习。
* EAS 系统具备一定的业务复杂度，既适合战略设计阶段，又适合战术阶段。

GitChat：您提到这次的 DDD 系列课程分为《领域驱动 **战略** 设计实践》和《领域驱动 **战术** 设计实践》两部分，这两个课程在内容设计上侧重有什么不同？很多读者关心《领域驱动 **战术** 设计实践》何时发布，可否透露一下？

张逸：这两个课程对应于 DDD 的战略设计阶段与战术设计阶段，粗略地说，前者更偏向于架构，后者更偏向于设计与编码。事实上，就我个人的规划来说，计划还有 **第三部分** ，是围绕着函数式编程讲解与 DDD 有关的实践，包括 EDA、CQRS、Domain Event 等知识。

目前，《领域驱动 **战略** 设计实践》还有最后几个章节没有完成。一旦完成后，就可以开始撰写《领域驱动 **战术** 设计实践》内容了。当然，战术设计的相关内容已有部分初稿，我争取能够在 11 月发布这部分内容。

GitChat：您觉得这门课的学员/读者应该是什么样的人？对于这些人，要想掌握领域驱动设计乃至在专业领域更上一层楼，您有哪些学习建议？

张逸：学习课程的学员/读者最好要有一定的软件设计能力，并对 DDD 学习抱有好奇心，希望能够将 DDD 学以致用。

学习建议：

* 积累领域知识，以提高沟通与协作能力；
* 以 Eric Evans 的《领域驱动设计》为主体，广泛涉猎与 DDD 相关的书籍与文章，并关注 DDD 社区的最新知识；
* 要善于总结，理清 DDD 中各个概念之间的区别与应用场景。

GitChat：作为一位曾就职于中兴、惠普、中软、ThoughtWorks 等大型中外企业的架构师/技术总监/首席咨询师，在职业发展方面，您对您的读者们有哪些建议？

张逸：我之前在 ThoughtWorks 的同事郑晔（校长）给我提过一个建议，就是打造自己的技术标签。例如，现在 DDD 就成为了我其中的一个技术标签了。这个说法的内在含义，就是要寻找和定位自己的技术发展方向，然后往更深的方向钻研，最终成为这个方向的技术专家。因此，结合自己的能力特长、兴趣点以及技术发展趋势去规划自己的技术发展方向，才是技术人员最应该思考并践行的。

到这里，访谈录的内容就结束了，大家若对 DDD 的内容比较感兴趣的话，欢迎订阅本课程，有任何关于本课程内容的疑惑可在读者圈留言给张逸老师，张老师会很积极的解答每一个问题~

[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

互动评论



评论



江文奇8 个月前

如果站在老板的角度 谁能回答为什么要使用领域设计

[鼓掌回复](javascript:void(0))

评论



英雄不转身8 个月前

领域就是某个专业方向，微服务也是为某个专业做服务

[鼓掌回复](javascript:void(0))

评论

查看更多

## 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