## 理解上下文映射

理解上下文映射

朗读人: 东急之花00′00′′ | 5.3M

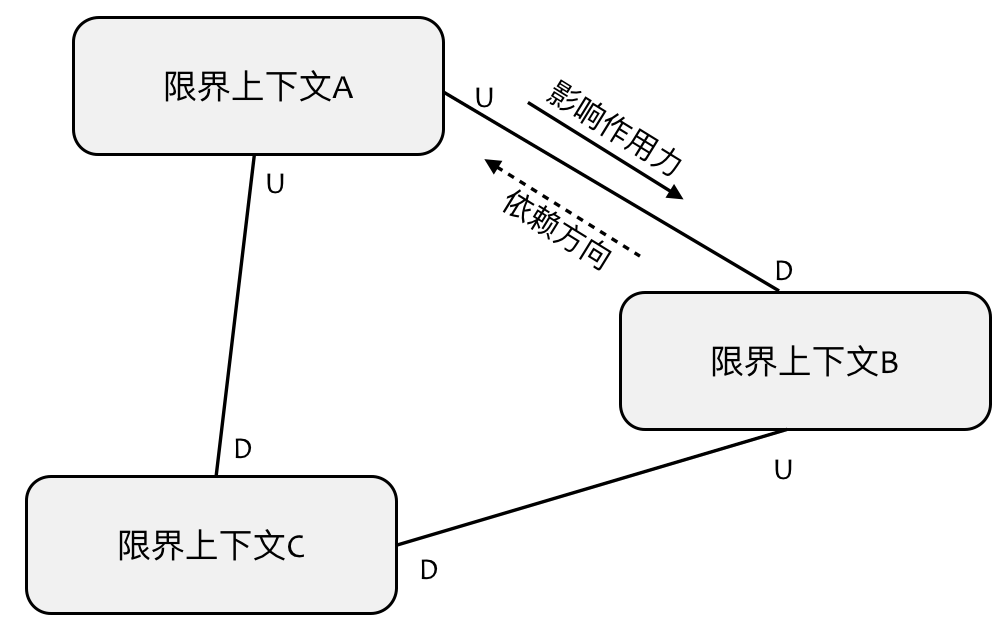
一个软件系统通常被分为多个限界上下文，这是运用“分而治之”思想来降低业务复杂度的有效手段，设计的难题往往会停留在“如何分”，然而限界上下文之间的“怎么合”问题同样值得关注，分与合遵循的还是软件设计的最高原则—— **高内聚、松耦合** 。分是合的基础，基于内聚相关度进行合理的分配，可以在一定程度减少限界上下文之间不必要的关联。假设分配是合理的，则接下来的“合”就是要尽可能地降低彼此之间的耦合。

既然前面提及限界上下文的识别是一个迭代过程，当我们在思考限界上下文该如何协作时，倘若发现协作总有不合理之处，就可能会是一个“设计坏味道”的信号，它告诉我们：之前识别的限界上下文或有不妥，由是可以审视之前的设计，进而演进为更为准确的限界上下文划分。即使抛开对设计的促进作用，思考限界上下文是如何协作的，仍然格外重要，我们既要小心翼翼地维护限界上下文的边界，又需要它们彼此之间良好的协作，并思考协作的具体实现方式，这个思考过程既牵涉到逻辑架构层面，又与物理架构有关，足以引起我们的重视。

领域驱动设计通过 **上下文映射（Context Map）** 来讨论限界上下文之间的协作问题，上下文映射是一种设计手段，Eric Evans 总结了诸如共享内核（Shared Kernel）、防腐层（Anticorruption Layer）、开放主机服务（Open Host Service）等多种模式。由于上下文映射本质上是与限界上下文一脉相承的，因此要掌握这些协作模式，应该从限界上下文的角度进行理解，着眼点还是在于“ **边界** ”。领域驱动设计认为：上下文映射是用于将限界上下文边界变得更清晰的重要工具。所以当我们正在为一些限界上下文的边界划分而左右为难时，不妨先放一放，在定下初步的限界上下文后，通过绘制上下文映射来检验，或许会有意外收获。

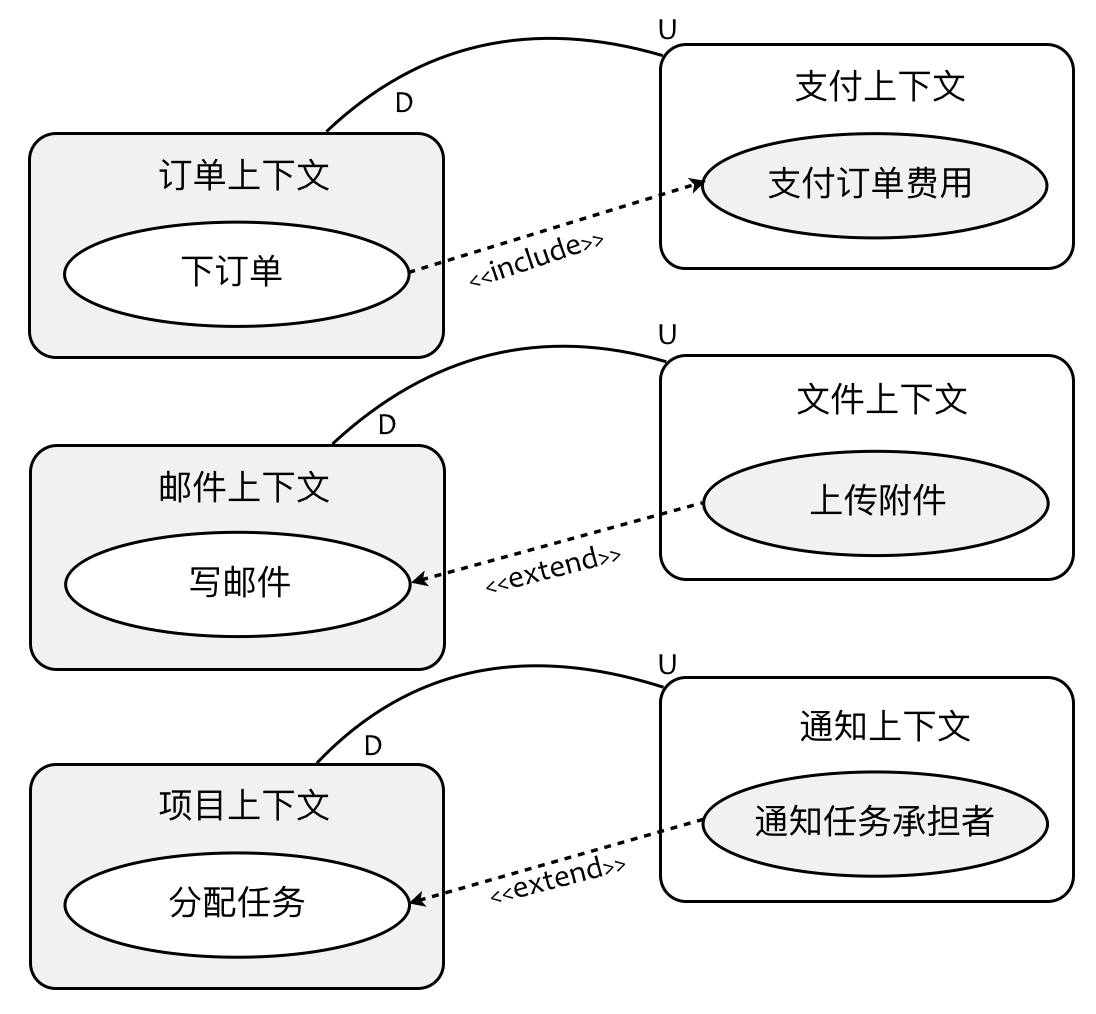
限界上下文的一个核心价值，就是利用边界来约束不同上下文的领域模型，以保证模型的一致性。然而，每个限界上下文都不是独立存在的，多数时候，都需要多个限界上下文通力协作，才能完成一个完整的用例场景。例如，客户之于商品、商品之于订单、订单之于支付，贯穿起来才能完成“购买商品”的核心流程。

两个限界上下文之间的关系是有方向的，领域驱动设计使用两个专门的术语来表述它们：“上游（Upstream）”和“下游（Downstream）”，在上下文映射图中，以 U 代表上游，D 代表下游，理解它们之间的关系，正如理解该术语隐喻的河流，自然是上游产生的变化会影响到下游，反之则不然。故而从上游到下游的关系方向，代表了影响产生的作用力，影响作用力的方向与程序员惯常理解的依赖方向恰恰相反，上游影响了下游，意味着下游依赖于上游。



enter image description here

在划分限界上下文的业务边界时，我们常常从“语义相关性”与“功能相关性”两个角度去判别职责划分的合理性。在上下文映射中，我发现之所以两个业务边界的限界上下文能产生上下游协作关系，皆源于二者的功能相关性，这种功能相关存在主次之分，往往是上游限界上下文作为下游限界上下文的功能支撑，这就意味着在当前的协作关系下，下游限界上下文中的用例才是核心领域。例如，订单与支付，下订单用例才是核心功能，支付功能作为支撑的公开服务而被调用；例如，邮件与文件共享，写邮件用例才是核心功能，上传附件作为支撑的公开服务而被调用；例如，项目管理与通知，分配任务用例才是核心功能，通知功能作为支撑的公开服务而被调用。巧的是，这种主次功能的调用关系，几乎对应的就是用例图中的包含用例或扩展用例。



enter image description here

如果我们通过用例图来帮助识别限界上下文，那么， **用例图中的包含用例或扩展用例或许是一个不错的判断上下文协作关系的切入点** 。选择从包含或扩展关系切入，既可能确定了职责分离的逻辑边界，又可以确定协作关系的方向，这就是用例对领域驱动设计的价值所在了。

那么，如何将上下文映射运用到领域驱动的战略设计阶段？Eric Evans 为我们总结了常用的上下文映射模式。为了更好地理解这些模式，结合限界上下文对边界的控制力，再根据这些模式的本质，我将这些上下文映射模式分为了两大类：团队协作模式与通信集成模式。前者对应的其实是团队合作的工作边界，后者则从应用边界的角度分析了限界上下文之间应该如何进行通信才能提升设计质量。针对通信集成模式，结合领域驱动设计社区的技术发展，在原有上下文映射模式基础上，增加了发布/订阅事件模式。

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