



链滴

领域驱动设计 DDD 之上下文映射图

作者: [valarchie](#)

原文链接: <https://ld246.com/article/1575041036213>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

上下文映射图的英文是Context Map其实这个翻译挺难理解的，上下文映射图其实就是不同上下文是何进行交流的关系。由于上下文映射图内容比较少。以下内容摘自《领域驱动设计精粹》。

三种集成方式

1. RPC方式
2. 消息队列或者发布-订阅机制。
3. RESTful方式

上下文映射的种类

- 合作关系：

合作关系存在于两个团队之间。每个团队各自负责一个限界上下文。两个团队通过互相依赖的一整套标联合起来形成合作关系。一损俱损，一荣俱荣。由于相互之间的联系非常紧密，他们经常会对同步程安排和相互关联的工作。他们还必须使用持续集成对保持集成工作协调一致。

- 共享内核：

两个或者多个团队之间共享着一个小规模但却通用的模型。团队必须就要共享的模型元素达成一致。可能他们当中只有一个团队会维护，构建及测试共享模型的代码。

- 客户-供应商：

两个限界上下文中，一方是供应商处于上游，一方是客户方处于下游。支配这种关系的是供应商，因为它必须提供客户需要的东西。客户需要与供应商共同制订规划来满足各种预期，但最终却还是由供应来决定客户获得的是什么以及何时获得。

- 跟随者：

上游团队没有任何动机去满足下游团队的具体需求。由于各种原因，下游团队也无法投入资源去翻译游模型的通用语言来适应自己的特定需求，因此只能顺应上游的模型。例如当一个团队需要与一个非庞大复杂的模型集成，而且这个模型已经非常成熟时，团队往往会成为它的跟随者。

- 防腐层：

这是最具防御性的上下文映射关系，下游团队在其通用语言（模型）和位于它上游的通用语言（模型）之间创建了一个翻译层。防腐层隔离了下游模型与上游模型，并完成了两者之间的翻译。所以，这也是一种集成方式。

- 开放主机服务：

开放主机服务会定义一套协议或者接口，让限界上下文可以被当做一组服务访问。该协议是开放的，有需要与限界上下文进行集成的客户端都可以相对轻松地使用它。通过应用程序编程接口提供的服务有详细的文档，用起来也很舒服。

个人认为比较好的方式是共享内核、防腐层、开放主机等形式的上下文映射。

关于DDD的理解各有不同，欢迎网友评论一起探讨。