## 运用领域场景分析提炼领域知识（上）

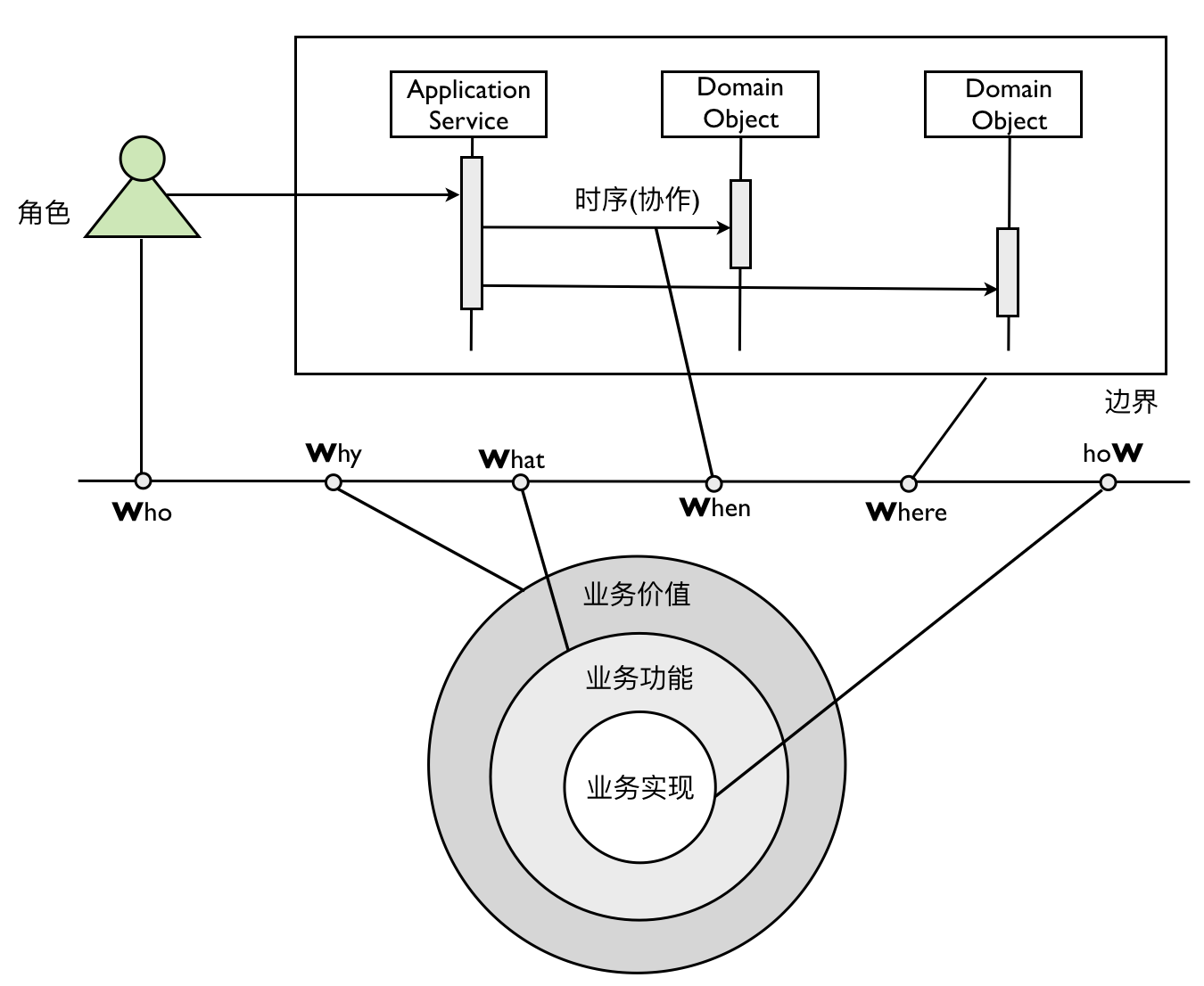
运用领域场景分析提炼领域知识（上）

朗读人: 东急之花00′00′′ | 4.1M

### 领域场景分析的 6W 模型

在软件构造过程中，我们必须正确地理解领域，一种生动的方式是通过“场景”来展现领域逻辑。领域专家或业务分析师从领域中提炼出“场景”，就好像是从抽象的三维球体中，切割出具体可见的一片，然后以这一片场景为舞台，上演各种角色之间的悲欢离合。每个角色的行为皆在业务流程的指引下展开活动，并受到业务规则的约束。当我们在描述场景时，就好像在讲故事，又好似在拍电影。

组成场景的要素常常被称之为 **6W 模型** ，即描写场景的过程必须包含 **W** ho、 **W** hat、 **W** hy、 **W** here、 **W** hen 与 ho **W** 这六个要素，6W 模型如下图所示：



通过场景分析领域需求时，首先需要识别参与该场景的用户角色。我们可以为其建立用户画像（Persona），通过分析该用户的特征与属性来辨别该角色在整个场景中参与的活动。这意味着我们需要明确业务功能（What），思考这一功能给该角色能够带来什么样的业务价值（Why）。注意，这里所谓的“角色”是参差多态的，同一个用户在不同场景可能是完全不同的角色。例如，在电商系统中，倘若执行的是下订单功能，则角色就是买家；针对该订单发表评论，参与的角色就变成了评论者。

在 6W 模型中，我将领域功能划分为三个层次，即业务价值、业务功能和业务实现，我将其称之为“职责的层次”。定义为“职责（Responsibility）”才能够更好地体现它与角色之间的关系，即“角色履行了职责”。业务价值体现了职责存在的目的，即解释了该领域需求的 Why。只有提供了该职责，这个场景对于参与角色才是有价值的。为了满足业务价值，我们可以进一步剖析为了实现该价值需要哪些支撑功能，这些业务功能对应 6W 模型中的 What。进一步，我们对功能深入分析，就可以分析获得具体的业务实现。业务实现关注于如何去实现该业务价值，因而对应于 hoW。

在电商系统中，买家要购买商品，因而 **下订单** 这一职责是具有业务价值的。通过领域分析，结合职责的层次概念，我们就可以得到如下的职责分层结构：

* 下订单
  + 验证订单是否有效
    - 验证订单是否为空
    - 验证订单信息是否完整
    - 验证订单当前状态是否处于“待提交”状态
    - 验证订单提交者是否为合法用户
    - 验证商品库存量是否大于等于订单中的数量
  + 基于业务规则计算订单总价、优惠与配送费
    - 获取用户信息
    - 获取当前促销规则
    - 计算订单总价
    - 计算订单优惠
    - 计算商品配送费
  + 提交订单
    - 将订单项插入到数据表中
    - 将订单插入到数据表中
    - 更新订单状态为“待付款”
  + 更新购物车
    - 删除购物车中对应的商品
  + 发送通知
    - 给买家发送电子邮件，通知订单提交成功，等待付款

当我们获得这样的职责层次结构之后，就可以帮助我们更加细致地针对领域进行建模。在利用场景进行建模时，还要充分考虑场景的边界，即 6W 模型中的 Where。例如，在“下订单”的案例中，验证商品库存量的业务实现需要调用库存提供的接口，该功能属于下订单场景的边界之外。领域驱动设计引入了 **限界上下文（Bounded Context）** 来解决这一问题。

针对问题域提炼领域知识是一个空泛的概念，业务场景分析的 6W 模型给出了具有指导意义的约束，要求我们提炼的领域知识必须具备模型的六个要素，这就好比两位侃侃而谈的交谈者，因为有了确定的主题与话题边界，一场本来是漫无目的野鹤闲云似的闲聊就变成了一次深度交流的专题高端对话。6W 模型也是对领域逻辑的一种检验，如果提炼出来的领域逻辑缺乏部分要素，就有可能忽略一些重要的领域概念、规则与约束。这种缺失会对后续的领域建模直接产生影响。正本清源，按照领域场景分析的 6W 模型去分析领域逻辑，提炼领域知识，可以从一开始在一定程度上保证领域模型的完整性。

### 领域场景分析的方法

我发现许多主流的领域分析方法都满足领域场景分析的 6W 模型，如果将 6W 模型看做是领域分析的抽象，那么这些领域分析方法就是对 6W 模型各种不同的实现。Eric Evans 在《领域驱动设计》一书中并没有给出提炼领域知识的方法，而是给出工程师与领域专家的对话模拟了这个过程。在领域驱动设计中， **团队与领域专家的对话必须是一种常态** ，但要让对话变得更加高效，使不同角色对相同业务的理解能够迅速达成一致，最佳的做法还是应该在团队中形成一种相对固定的场景分析模式，这些模式包括但不限于：

* 用例（Use Case）
* 用户故事（Use Story）
* 测试驱动开发（Test Driven Development）

#### 用例

用例（Use Case）的概念来自 Ivar Jacobson，它帮助我们思考参与系统活动的角色，即用例中所谓的“参与者（Actor）”，然后通过参与者的角度去思考为其提供“价值”的业务功能。Jacobson 认为：“用例是通过某部分功能来使用系统的一种具体的方式……因此，用例是相关事务的一个具体序列，参与者和系统以对话的方式执行这些事务。……从用户的观点来看，每个用例都是系统中一个完整序列的事件。”显然，用例很好地体现了参与者与系统的一种交互，并在这种交互中体现出完整的业务价值。

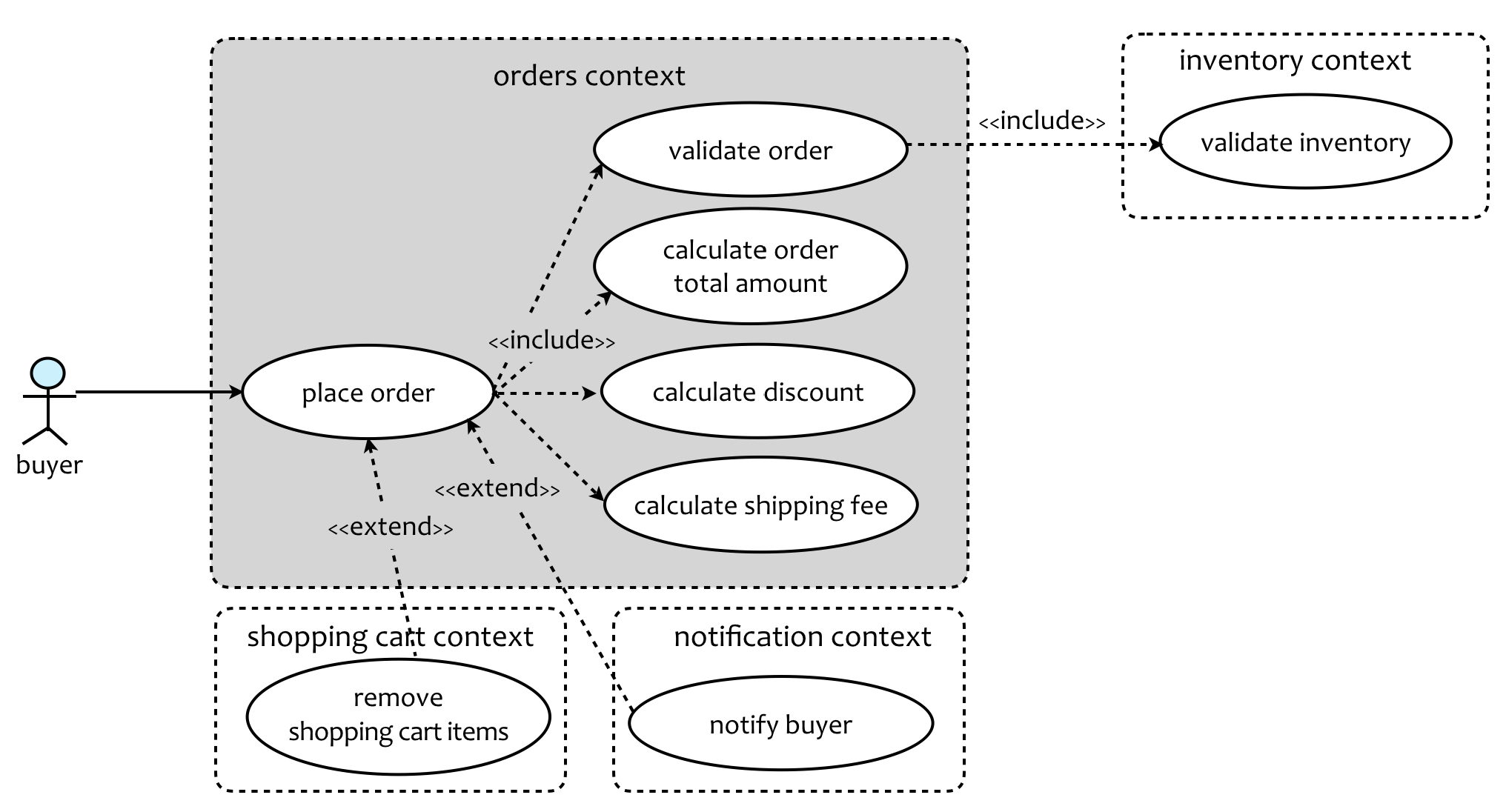
用例往往通过用例规格说明来展现这种参与者与系统的交互，详细说明该用例的顺序流程。例如，针对“买家下订单”这个用例，编写的用例规格说明如下所示：

用例名称：买家下订单  
用例目的：本用例为买家提供了购买心仪商品的功能。  
参与者：买家  
前置条件：买家已经登录并将自己心仪的商品添加到了购物车。  
  
基础流程：  
1. 买家打开购物车  
2. 买家提交订单  
3. 验证订单是否有效  
4. 计算订单总价  
5. 计算订单优惠  
6. 计算配送费  
7. 系统提交订单  
8. 删除购物车中对应的商品  
9. 系统通过电子邮件将订单信息发送给买家  
  
替代流程：系统验证订单无效  
在第3步，系统确认订单无效，提示验证失败原因  
  
替代流程：提交订单失败  
在第7步，系统提交订单失败，提示订单失败原因

虽然文本描述的用例规格说明会更容易地被业务分析人员和开发人员使用和共享，但是这种文本描述的形式其可读性较差，尤其是针对异常流程较多的复杂场景，非常不直观。UML 引入了用例图来表示用例，它是用例的一种模型抽象，通过可视化的方式来表示参与者与用例之间的交互，用例与用例之间的关系以及系统的边界。组成一个用例图的要素包括：

* 参与者（Actor）：代表了 6W 模型的 **W** ho；
* 用例（Use Case）：代表了 6W 模型的 **W** hat；
* 用例关系：包括使用、包含、扩展、泛化、特化等关系，其中使用（use）关系代表了 **W** hy；
* 边界（Boundary）：代表了 6W 模型的 **W** here。

通过用例图来表示上面的用例规格说明：



enter image description here

在这个用例图中，为什么只有 place order 用例与 buyer 参与者之间才存在使用（use）关系？我们可以看看上图中的所有用例，只有“下订单”本身对于买家而言才具有业务价值，也是买家“参与”该业务场景的主要目的。因此，我们可以将该用例视为体现这个领域场景的主用例，其他用例则是与该主用例产生协作关系的子用例。

用例之间的协作关系主要分为两种：

* 包含（include）
* 扩展（extend）

如何理解包含与扩展之间的区别？大体而言，“包含”关系意味着子用例是主用例中不可缺少的一个执行步骤，如果缺少了该子用例，主用例可能会变得不完整。“扩展”子用例是对主用例的一种补充或强化，即使没有该扩展用例，对主用例也不会产生直接影响，主用例自身仍然是完整的。倘若熟悉面向对象设计与分析方法，可以将“包含”关系类比为对象之间的组合关系，如汽车与轮胎，是一种 must have 关系，而“扩展”关系就是对象之间的聚合关系，如汽车与车载音响，是一种 nice to have 关系。当然，在绘制用例图时，倘若实在无法分辨某个用例究竟是包含还是扩展，那就“跟着感觉走”吧，这种设计决策并非生死攸关的重大决定，即使辨别错误，几乎也不会影响到最后的设计。

无论是包含还是扩展，这些子用例都是为主用例服务，体现了用例规格描述的流程，即为 6W 模型中的 **W** hen 与 ho **W** 。

根据用例代表的职责相关性，我们可以对用例图中的所有用例进行分类，从而划分用例的边界。确定用例相关性就是分析何谓内聚的职责，是根据关系的亲密程度来判断的。显然，上图中的 remove shopping cart items、notify buyer 与 validate inventory 与 place order 用例的关系，远不如 validate order 等用例与 place order 之间的关系紧密。因此，我们将这些用例与 order 分开，分别放到 shopping cart、notification 与 inventory 中，这是用例边界（ **W** here）的体现。

用例图是领域专家与开发团队可以进行沟通的一种可视化手段，简单形象，还可以避免从一开始就陷入到技术细节中—— **用例的关注点就是领域** 。

绘制用例图时，切忌闭门造车，最好让团队一起协作。 **用例表达的领域概念必须精准** ！在为每个用例进行命名时，我们都 **应该采纳统一语言中的概念** ，然后以言简意赅的动宾短语描述用例，并提供英文表达。很多时候，在团队内部已经形成了中文概念的固有印象，一旦翻译成英文，就可能呈现百花齐放的面貌，这就破坏了“统一语言”。为保证用例描述的精准性，可以考虑引入“局外人”对用例提问，局外人不了解业务，任何领域概念对他而言可能都是陌生的。通过不断对用例表达的概念进行提问，团队成员就会在不断的阐释中形成更加清晰的术语定义，对领域行为的认识也会更加精确。

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