



链滴

领域驱动设计 DDD 之领域服务

作者: [valarchie](#)

原文链接: <https://ld246.com/article/1573866923108>

来源网站: [链滴](#)

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

什么是领域服务?

在战术建模当中，并非所有模型都是事物。有些模型是对领域中的一些行为操作进行建模。此类我们称之为领域服务。当一些重要的领域操作无法放到实体、值对象或者聚合时，他们本质是行为不是事物。如果我们不寻找一些对象来封装这些领域行为的话，又会演变成之前过程式的编程方式。希望在领域设计当中统一用模型对象进行交互。此时领域服务使用细粒度的领域对象如实体或者值对象进行交互，在服务内部描述领域知识得出结果并将其返回。

<blockquote>

<p>领域服务的参数和返回类型应该是领域对象。</p>

</blockquote>

三个特征:

它是与领域相关的操作如执行一个显著的业务操作过程，但它又并不适合放入实体与值对象中。

操作是无状态的。

对领域对象进行转换，或以多个领域对象作为输入进行计算，结果产生一个值对象。

<p>总结:

当领域中的某个重要的过程或者转换操作不是实体和值对象的自然职责时，应该在模型中添加一个作独立接口的操作，并将其声明为领域服务。定义接口时要使用模式语言，并确保操作名称是业务统一用语言的术语。除此之外领域服务要成为无状态的。

<blockquote>

<p>我们需要把握一个度，如果将所有领域知识都封装到实体当中去时，模型就会因为行为过多导致充血，充血情况下很多实体的行为都是勉强且不自然的。而让我们将行为一股脑的封装到领域服务当中的话，此时模型就因毫无行为导致贫血，我们又回到了传统开发模式了。</p>

</blockquote>

区分不同的服务

在传统的开发中我们已经有 Service 服务的概念了，这时候再引入领域服务时，我们可能就会混淆。在领域驱动设计中我们主要将服务分为三类，一类是应用服务，一类是领域服务，一类是基础服务。如何去区分这三种服务呢？我的一个简单的理解是通过服务自身所服务的客户端来进行区分。应用服务提供面向用户的服务，它所完成的是一整个用户需求。领域服务提供面向应用层的服务，它所完成的是封装领域知识，供应用层使用。基础服务提供面向应用层和领域层的服务，它所提供的是项目中个层都可能使用到的通用功能。

<p>我们举一个银行转账的例子，通过不同服务所处理的事情来说明</p>

应用服务：获取输入，发送消息给领域层，监听确认消息，决定使用基础服务来发送邮件。

领域服务：协调账户模型和总账模型进行交互，执行相应的领域行为。

基础服务：按照应用服务的指示发送邮件。

粒度

应用层负责对领域对象的行为进行协调，来满足某一个领域需求。但如果领域对象都是实体和值对象等细粒度的对象时，应用层就得去了解这些细粒度的对象如何交互才能满足领域需求。那此时便将域内的知识泄露到应用层了。我们应该尽量避免领域知识泄露到应用层当中去。那此时领域服务就不为一种良好的处理方式，通过将细粒度的领域对象封装到领域服务当中去，将领域知识限制在领域服当中，形成粗粒度的领域对象。因此应用层调用粗粒度的领域服务时也就无需关心其中复杂的对象交互。

转换过程

我们举一个商铺系统中需要统计每日台账的功能，而这个功能我们放置到台账统计服务当中去。们通过传入一个商铺实体以及日期，在方法内使用 Repository 资源库（同 DAO），进行获取领域模型，再利用领域模型间的交互来完成台账的统计，最后封装成一个台账模型返回出去。

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">public class LedgerAccountor {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    public LedgerSummary ledgerSummary(Shop aShop, Date date){
```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> int totalInco
e = 0;
</span></span><span class="highlight-line"><span class="highlight-cl"> int totalExpe
se = 0;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> List<ShopT
ade> trades = ShopTradeRepository.getShopTrades(ashop,date);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> for(ShopTrad
trade : trades){
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> totalInco
e += trade.getIncome();
</span></span><span class="highlight-line"><span class="highlight-cl"> totalExpen
e += trade.getExpense();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> return new L
dgerSummary(totalIncome, totalExpense);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<h5 id="关于DDD的理解各有不同-欢迎网友评论一起探讨-">关于 DDD 的理解各有不同，欢迎网
评论一起探讨。</h5>

```