

基于设计模式对 WV 网关业务逻辑处理模块的重构^①

Refactoring the Service Logic Module of WV Gateway Based on Design Patterns

张 婧^{1,2}, 朱晓民^{1,2}, 廖建新^{1,2}

(1. 北京邮电大学 网络与交换技术国家重点实验室 北京 100876;

2. 东信北邮信息技术有限公司 北京 100083)

摘 要: 为了使 WV (Wireless Village) 网关系统的实现更加松耦合, 便于今后更容易地进行功能或者性能的改进, 基于面向对象设计中的“开-闭”原则, 合成/聚合复用原则和依赖倒转原则等重要原则, 对系统中的业务逻辑处理模块进行了重构, 使用了一些经典的行为模式和结构模式, 如: 策略 (Strategy) 模式、代理 (Proxy) 模式、桥梁 (Bridge) 模式等。

关键词: WV 网关 设计模式 重构

WV (Wireless Village) 网关是中国移动飞信系统的一部分, 用于实现 WV 协议与 SIP - C (Compact SIP) 协议之间的转换^[1], 使得支持 WV 协议的手机 IM (Instant Message) 客户端能够接入到使用 SIP - C 协议的 IM 服务器^[2], 与 SIP - C 用户和互联网用户实现互通。2001 年 4 月 WV 协议由 Nokia、Motorola、Ericsson 三大移动电话厂商提出, 而后 OMA (Open Mobile Alliance) 组织陆续制定和发布了 IMPS (Instant Message and Presence Service) 1.1, 1.2, 1.3 版本^[3-5]。业务逻辑处理模块实现了基本的 WV 网关业务, 通过必要的适配后与 SIP - C 服务器交互以提供即时消息、Presence 状态、联系人、群组等功能。

本文主要是基于设计模式对业务逻辑处理模块的重构, 设计模式是面向对象设计中很有价值的经验总结, 在软件设计中合理的使用设计模式, 可大大增强系统的开放性、兼容性、稳定性和可扩充性。

1 引言

1.1 文章安排

第 2 节介绍相关工作。第 3 节分别介绍三个地方

的重构, 主要是将代理模式用于实现群组权限判断与群组功能的分离, 将策略模式用于通用搜索功能设计, 将桥梁模式用于整个业务逻辑调用数据库操作, 在每一个应用中阐述了原有实现方案的缺点和选择新模式的原因及优点等。第 4 节给出结论。

2 相关工作

WV 网关的整体架构采用了 MVC (Model View Controller) 框架, 如图 1 所示。本文的重构就是针对业务处理逻辑模块进行的, 在恰当的地方使用合适的设计模式, 重新设计, 改变程序的内部结构。设计模式可以看成是前人对一些代表性问题的代表性解决方案。每一种模式都使用了若干面向对象设计的原则, 策略模式属于对象的行为模式, 针对一组算法, 将每一个算法独立封装在一个类中, 由客户端决定使用哪个算法, 将行为和环境分离开来。代理模式属于对象的结构模式, 给某一个对象提供一个代理对象, 并由代理对象控制队员对象的引用。桥梁模式主要用于将抽象化与实现化脱耦, 使得二者可以独立地变化^[6]。

^① 基金项目: 国家杰出青年科学基金 (No. 60525110), 国家 973 计划项目 (No. 2007CB307100, 2007CB307103), 新世纪优秀人才支持计划 (No. NCET-04-0111), 电子信息产业发展基金项目 (基于 3G 的移动业务应用系统)

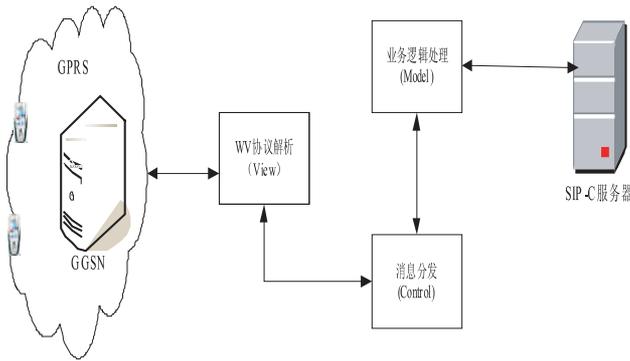


图 1 总体示意图

3 基于设计模式的重构

3.1 群组权限判断与群组功能实现相分离的设计

IMPS 规定的群组功能,分为创建群组、删除群组、加入群组、离开群组、群组成员管理、修改群组属性、拒绝用户加入群组、订阅群组变化八个部分。面对群组功能,WV 用户有不同的权限管理,IMPS1.2 将用户分为会员和非会员两种基本类型^[4],而只有一个群组的管理者才有权授予某个 WV 用户的会员资格,同时只有会员才有资格加入群组,已经加入群组的会员又分为管理者、协调者、普通用户三种权限级别。而 IMPS 1.1 中没有会员资格这层权限。不同的规范版本下,群组的功能处理逻辑是一样的,而权限管理策略却不同。根据封装变化的原则,这里需要将权限管理和群组功能相分离。另一方面可以借用 AOP(Aspect - Oriented Programming)的思想,把权限管理提出来形成一个单独的纵面。

WV 网关系统中,原来通过装饰模式来实现权限管理和群组功能的分离,具体类结构如图 2。

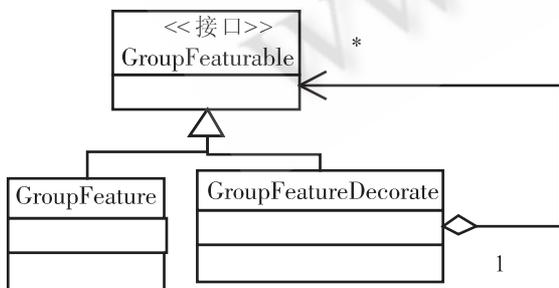


图 2 原类图

接口 GroupFeaturable 定义了群组操作的所有接

口,类 GroupFeature 实现了群组的基本操作,类 GroupFeatureDecorator 中的每一个方法都是首先对提交请求的用户进行权限判断,如果有权限再调用类 GroupFeature 中相应的方法来具体处理。这种情况下,调用者需要创建原功能类,将原功能类传递给新功能类的构造函数。比如,实现删除群组的功能,调用者需要写如下代码:

```
GroupFeaturable groupFeature = new GroupFeature();
GroupFeaturable groupFeatureDecorate = new GroupFeatureDecorator(groupFeature);
groupFeatureDecorate.deleteGroup( userId, groupId );
```

装饰模式强调能以对客户端完全透明的方式对原有类进行包装,增加一些功能,这里将权限管理看成是对原有功能的扩展,我们觉得有些牵强,不符合使用装饰模式的通常情况,也没有充分体现装饰模式的优点。而这里的需求是为了对权限的管理与具体群组功能处理相分离,看起来更符合代理模式使用的情况,代理模式的主要特性就是给某一个对象提供一个代理对象,并由代理对象控制对原对象的引用,这里的权限管理正是需要代理对象出现来决定是否可以引用原对象。基于代理模式实现的类图如图 3:

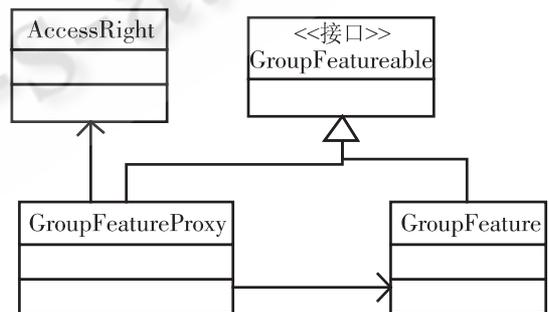


图 3 新类图

由于代理模式下代理类与原功能类之间有了委任关系,GroupFeature 是 GroupFeatureProxy 的成员变量,因此实现删除群组的功能,只需要写如下代码:

```
GroupFeaturable groupFeatureProxy = new GroupFeatureProxy();
groupFeatureProxy.deleteGroup( userId, groupId );
```

与原来装饰模式下的代码相比,调用者只需要创建一个代理对象,代理对象的删除方法中首先调用一个权限策略对象进行权限判断,在请求者有权限的情况下,再调用群组功能对象中删除群组的方法来完成删除群组的功能。针对不同的规范版本下权限管理策略的不同,代理分别调用不同的权限策略对象就可以。客户端不知道代理对象的存在,以为在跟原有功能对象交互,这也更加符合“只与朋友谈话”的原则^[6]。

3.2 通用搜索功能的设计

WV 用户可以搜索符合一定条件的用户和群组,不同的规范版本定义的搜索策略不尽相同,原来网关只支持基于 private profile,手机号条件搜索用户的功能,设计如图 4。现在需要增加基于 public-profile 和群组属性等的搜索策略^[5],同时考虑到以后更加方便的增加或者修改不同的搜索策略,将策略模式引入了该功能模块设计中,因为策略模式的主要特点就是能将行为和环境分开,这样环境和客户端都不会受到算法的增减、修改的影响。

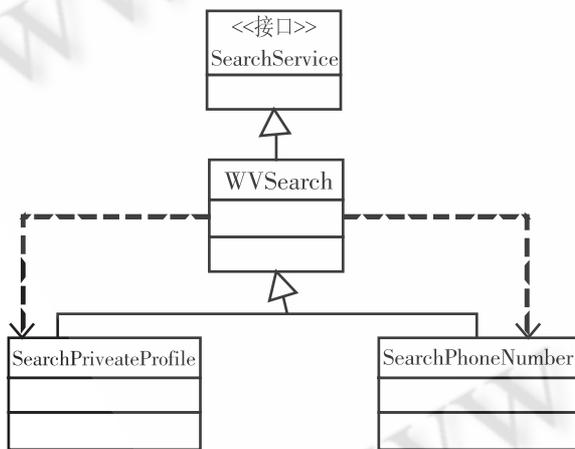


图 4 原类图

SearchService 定义了进行搜索的所有接口, WVSearch 实现公有的方法,而 SearchPrivateProfile 和 SearchPhoneNumber 继承 WVSearch,实现 SearchService 定义的其他接口,同时 WVSearch 根据不同的请求类型创建不同的具体搜索策略类实例。这种设计方案下,如果增加新的搜索策略,需要生成继承 WVSearch 的搜索策略类,还需要修改 WVSearch,这违背了对扩展开放,对修改封闭的原则^[7];另一方面继承的层次

多,定义的 SearchService 接口和 WVSearch 抽象类意义重叠,应该倾向于用复合代替继承。针对以上两点缺陷,采用了以下的设计方案:

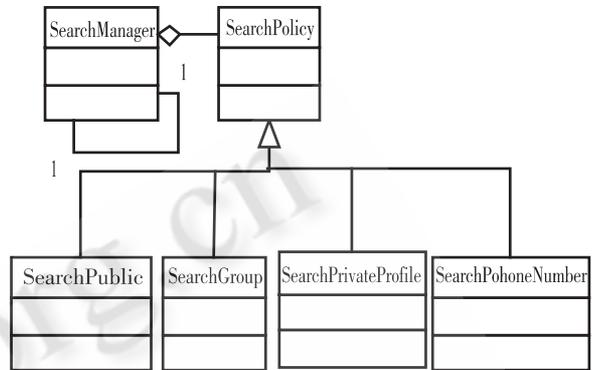


图 5 新类图

SearchPolicy 是个抽象类,实现了公有方法,给每一个搜索请求分配搜索 ID,在数据库中添加或者提取搜索结果等,同时定义了抽象的搜索策略方法,具体由不同的策略类实现。SearchManager 接受客户端提交的搜索请求,持有抽象策略类的引用,可以根据搜索请求调用一个具体的搜索策略对象。考虑到具体搜索策略以后可能会增加或者减少,这里增加了 SearchManager 类,根据不同的请求类型创建不同的具体搜索策略类实例,以实现封装变化原则。SearchManager 相当于一个工具类,因此使用了单例模式,保证了 WV 网关系统中只有一个 SearchManager 实例来处理所有客户端提交上来的搜索请求。

3.3 业务逻辑处理中调用数据库和支持不同版本的商业逻辑处理的设计

在业务逻辑处理中需要对数据库中的有关数据进行必要的增删改查,比如 联系人列表和联系人的相关信息都存在数据库中,在进行联系人列表管理时需要与数据库交互。在原来设计中,商业逻辑类 WVservletProcessor 与数据库操作 TorqueDAO 是复合关系,如图 6:

考虑到数据库操作具体实现方式可能会变化,例如,原来采用开源的持久层工具 Torque,重构时需要用 Hibernate 替换掉 Torque,因为 Hibernate 支持二级缓存,性能上更优化,所以将涉及到的数据库操作抽象成了一个 WVStoreDAO 接口。在这个地方使用了面向接

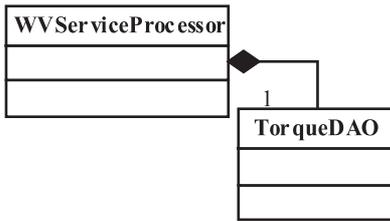


图 6 原类图

口编程的原则,实现松耦合,将依赖于具体类的地方改为面向接口,也就是说具体数据库操作类的改变将不会影响到商业逻辑类的处理。

同时考虑到 WV 规范 1.3 在一些功能点上进行的逻辑处理与 WV 规范 1.2 和 1.1 有所不同,这样就需要添加一个新的 processor 来实现,而对于同一个功能点的不同的 processor 中逻辑处理的部分又存在相同的地方,比如:处理的步骤一样,就可以将相同的操作提到抽象类中,应用模版模式。

这样,就建立起两个继承体系,通过委托实现一个对另一个的调用^[8]。这就是通过桥梁模式来实现业务逻辑处理中调用数据库和支持不同版本的商业逻辑处理,类结构如图 7。重构后,系统的抽象化角色是业务逻辑处理,代表系统实现化角色则是数据库操作,如果系统中需要加入新的 DAO 类型或者 WVServiceImpl 类型的话,系统中已有的各个角色不必改变,只需要改变一个多态性的聚合关系。这个设计通过使用组合/聚合复用原则来达到“开-闭”原则要求的。

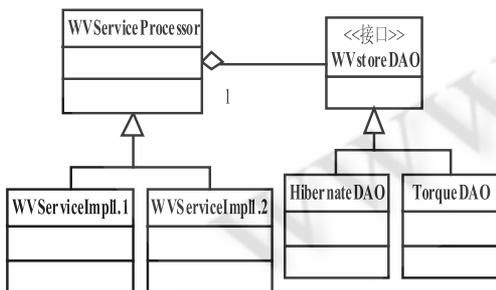


图 7 新类图

4 总结

本文主要讨论了对业务逻辑处理模块中三个方面的重构,为在重构中灵活运用设计模式进行了一些探索,或者改变原来不合适的设计模式,或者根据需求使用了一些经典的设计模式,使得业务逻辑处理模块的结构更加松耦合,便于今后 WV 网关的开发和维护。

参考文献

- 1 彭家恒,廖建新,王纯,阮友森. 基于 MVC 的 WV 网关系统的设计与实现. 北京工商大学学报(自然科学版),2007(2):38-41.
- 2 袁琪,廖建新,朱晓民,杨军. 即时消息系统中 WV 网关的设计与实现. 计算机系统应用,2007,16(3):63-66.
- 3 OMA. OMA - WV - CSP - V1_1 - A. 2002 - 10 - 01. 12.
- 4 OMA. OMA - IMPS - WV - CSP - V1_2 - A. 2005 - 01 - 25. 71 - 79.
- 5 OMA. OMA - TS - IMPS_CSP - V1_3 - A. 2007 - 01 - 23. 82 - 86.
- 6 阎宏. Java 与模式. 北京:电子工业出版社. 2002:621-623.
- 7 Freeman E, Freeman E, Bates B, Sierra K. Head First Design Patterns. California: O'Reilly. 2004. 105.
- 8 Martin Fowler 著,侯捷,熊节译. 重构 - 改善既有代码的设计.(第 1.1 版.)北京:中国电力出版社. 2003. 362-367.