

多媒体彩铃项目中 UP 框架的设计与应用^①

陈亮, 朱晓民

(北京邮电大学 网络与交换技术国家重点实验室, 北京 100876)

(东信北邮信息技术有限公司, 北京 100191)

摘要: 设计了一种针对多媒体彩铃项目 Portal 系统的统一开发框架, 并举例说明该框架在多媒体彩铃业务中的应用。UP(Unify Portal)框架基于传统的 Spring + Hibernate 开发框架, 延续并优化了分层的体系结构; 使用了面向接口编程和 AOP(Aspect Oriented Programming, 面向切面编程)的设计思想; 统一了 Portal 系统的设计理念和开发方式。在开发中确定了数据的设计方式, 业务流程的组成模式及异常处理系统。运用该框架开发的 Portal 系统可以同时支持多种接入方式的功能需求, 并保证了高可重用性, 可维护性。同时 UP 框架是一种可以进行推广的面向大多数 Portal 系统开发的统一框架。

关键词: 多媒体彩铃; 开发框架; spring; hibernate; portal; AOP

Design and Application of Unify Portal Framework for MRBT

CHEN Liang, ZHU Xiao-Min

(State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

(EBUPT Information Technology Co. Ltd., Beijing 100191, China)

Abstract: In this paper, a new framework for MRBT(Multimedia Ring Back Tone) is presented, which is based on the traditional Spring + Hibernate development framework. It extends and optimizes the hierarchical architecture, uses the design of Interface-oriented programming and AOP (Aspect Oriented Programming) and unifies the development methods of portal system. This framework consists of identified data, reasonable business processes and exception handling system. Portal development using the framework can simultaneously support multiple access modes and ensure high reusability and maintainability. UP framework can also be promoted as a development framework for the most portal system.

Keywords: multimedia ring back tone; development framework; spring; hibernate; portal; AOP

多媒体彩铃(MRBT)业务是一种采用多种媒体作为回铃音的增值业务^[1]。作为对 2G 彩铃(CRBT)业务的扩展, 多媒体彩铃业务允许被叫用户为呼叫自己移动电话的其他用户设定特殊的媒体, 主叫用户不再是仅仅听到单调的“嘟?嘟?”振铃声或单纯的歌曲、乐曲等 2G 彩铃, 而是可以全方位感受极具个性的视频、动画、音乐、图片或文字等^[2]。该业务是通过在原有的通信网上叠加智能网, 通过智能网中的独立智能外设(本文称为多媒体彩铃平台)来实现^[3]。多媒体彩铃平

台提供多种接入方式, 包括 IVR (Interactive Voice Response, 交互语音应答系统), Web, WAP(Wireless Application Protocol, 无线应用协议), SMS(Short Messaging Service, 短消息服务)等方式^[4]。其中最主要的方式就是 Web 接入方式。这些接入方式可以统称为多媒体彩铃 Portal。Portal 是整个多媒体彩铃平台为用户提供服务的接口, 为各种用户提供不同方式的接入能力, 包括: 向最终用户提供使用彩铃业务的途径, 向运营商提供对彩铃业务数据的管理及维护功能。为

^① 基金项目:国家杰出青年科学基金(60525110);国家 973 计划(2007CB307100,2007CB307103);国家自然科学基金(60902051);中央高校基本科研业务费专项资金(BUPT2009RC0505);电子信息产业发展基金

收稿时间:2010-07-27;收到修改稿时间:2010-08-19

了给用户提供这些功能,Portal 需要拥有和数据库交互的能力,并保持正确的处理逻辑流程。多媒体彩铃平台中这部分功能被抽象成 Portal 系统,其中最重要的是 Portal 接口部分的设计和开发。

随着多媒体彩铃相关的业务不断增加,传统的 Portal 接口逐渐变得代码冗长、结构混乱,难以进行维护。同时,多媒体彩铃业务开展的状况是各省独立维护,各省业务千差万别。多媒体彩铃的多种接入手段如 WWW、SMS、WAP 和 IVR 之间的资源互相独立^[5],对于新业务的开发或者业务的修改升级带来了很大的压力和重复工作。

为了使我们在多媒体彩铃平台上的业务维持可持续发展,本文提出了统一 Portal 开发框架,将 IVR 语音管理、WWW 网站、SMS 短信业务以及开放接口的底层逻辑部分实现,进行整合,力求做到逻辑统一、代码清晰、可维护性强^[6]。同时也为 Portal 系统开发提供了一种新的开发模式。

1 原有框架的设计和问题

1.1 原有框架的设计

原有的数据结构设计如图 1 所示。由于多媒体彩铃业务在多个省份部署,不同省份相互独立,对同样一个数据元素比如铃声 Ring,要求有各种不同的形式。在原有的设计中针对每一个不同的 Portal 要求,我们 Portal 接口都需要提供不同的数据结构。

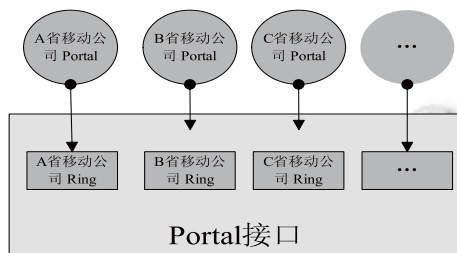


图 1 原多媒体彩铃 portal 数据设计图

原有的逻辑流程分层设计如图 2 所示。针对 Portal 的每一组接口,我们设定一个 Service 层,用于处理所有的判断和处理流程,而 Dao 层是与数据库直接交互的操作逻辑。两层的结构也是 Spring + Hibernate 经典框架的经典分层结构,它将业务逻辑和处理逻辑解耦,解决业务提供和数据处理耦合的问题。而且这样的设计容易理解,简单易学,但是也存在很多问题,这些

问题将在 1.2 节中具体分析。



图 2 原多媒体彩铃逻辑流程分层图

1.2 原有框架的问题

原有框架在设计上使用了最传统,最经典的开发框架,从某种程度上可以减少系统耦合度,增加代码可重用性。但在多媒体彩铃实际项目中,这种开发框架暴露出一系列问题,主要有以下几个方面。

数据异构性:从 1.1 节我们可以看到,不同省份所要求的数据形式并不相同,但又有相同的部分,而且都支持最基本的业务操作。然而由于各自的独立性,所以导致数据结构出现大量代码冗余和代码混乱。这样的结果就是开发人员和维护人员难以控制统一版本,各地代码出现不同步的现象。

流程异构性:各省之间除了数据不一致,业务流程也有着不同。同样一个开户流程,不同省份可能大部分操作相同,而同时有存在细小的差异。有些省份具有一些其他省份不具备的业务需求。1.1 节的两层流程设计并不能很好的解决这个问题。每个类似的接口内部都需要书写业务实现的流程,而这部分代码的更新和维护将会是一个繁琐而易错的部分。同样在开发中不同接口的开发也会浪费大量的人力。

异常处理不统一:异常的正确处理是软件保证正常运行的一个基本条件。原有的设计中,异常抛出代码穿插在业务逻辑中,属于重量型设计方式。这种模式下,异常的整个设计侵入到运行流程中来,这样不但可读性差,而且会使两者的管理也会趋于复杂。

基于以上分析,多媒体彩铃项目中,Portal 接口部分要求一种数据结构灵活,业务流程合理,异常处理规范的统一的 Portal 开发框架。本文提出的 UP(Unify Portal)开发框架可以很好的解决这些问题。

2 UP框架的设计

2.1 总体设计

针对原有框架的三个重大问题,并结合了 Spring + Hibernate 的底层接口设计思想,我们在原有开发框架中做出 3 个比较大的改动,从而形成一套统一的 Portal

接口开发框架。如图 3 所示，右侧的数据结构并没有平行的分成多个不同的类，而是拥有一定关系的两类数据 POJO(Plain Ordinary Java Objects)和 Portal Data。具体设计我们在 2.2 节讨论。在分层结构方面，我们在原有的 Service 层和 Dao 层之间添加了一个 Module 层，这一层的加入可以解决数据和流程异构的问题，具体我们在 2.3 节讨论。最后在 Module 层中我们分离出来一个 RuleModule 模块，该模块用于处理在业务逻辑中所有的异常，可以很好的解决异常处理不统一的问题，具体我们在 2.4 节讨论。

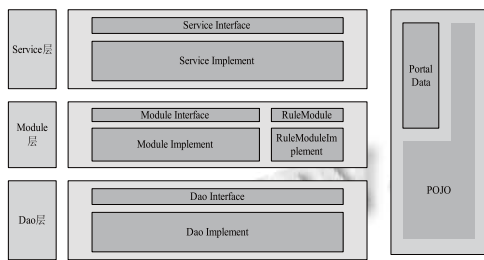


图 3 UP 框架总体设计图

2.2 数据设计

在多媒体彩铃项目中，数据是为了实现各省的业务功能而针对现实的对象而设定的对象。所以不同的业务功能对应不同的对象。这些现实的对象从不同的项目参与者角度看是不同的。例如同样的铃音资源，用户关心的是铃音的名称，歌手，价格和有效期；业务开发人员则关心铃音的播放文件位置，铃音状态等更多的信息。从不同的地域分，同样的对象也会有不同属性和需求。然而这些对象属于用一种现实的对象，从面向对象程序开发最基本原则上讲，这些对象拥有的共同属性应当抽象出来，而数据结构中的差异部分应该在抽象结构的基础上进行扩展。

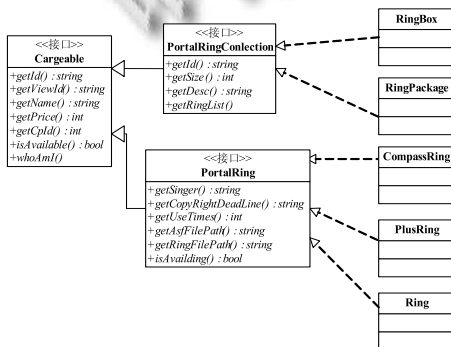


图 4 多媒体彩铃 Ring 数据设计图

在抽象和扩展的设计上一共有两种方法，继承和接口。继承是耦合性比较高的方式，对代码有一定的侵入性，所以我们选择接口的方式。从各种 POJO 对象中抽象出一个 Portal 所使用的通用接口，如果这些接口还可以抽象，就需要将这些接口再抽象出一个接口父类。以铃音资源为例，在多媒体彩铃项目中设计及实现方式如图 4 所示。

2.3 层次设计

正如 1.2 节我们讨论的一样，Service 层和 Dao 层无论从数据或是流程方面，都是耦合程度相当高的。从数据层次讲，Dao 层操作的是具体的实际数据形式，而 Service 层如果操作同样的数据形式，那么各省市数据形式之间的异构性将会无法解决。Portal 接口就不能作为一个统一的接口使用，这样将会使我们的接口设计丧失存在的意义。如果我们强制 Dao 层返回 Portal Date 的数据形式，从某种程度将会降低数据异构性，并使得 Service 对外提供统一的接口。但这样会有致命的问题——流程异构性。各省市之间由于业务属性和需求的不同，对于同一种操作可能有不同的实现方式。如果对每种操作都新开发一个接口，并在接口中使用类似的代码实现，这样会导致同样功能的代码出现在多个地方。这不仅是一种人员和时间上的浪费，更对后续维护和变更操作造成巨大隐患。既然两层结构已经不能满足现有的需求，层次的变更就势在必行了。在 2.1 节的 UP 框架总体设计中，添加的 Module 层可以完美的解决这一问题。具体的设计如图 5 所示。

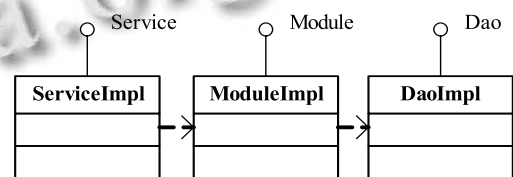


图 5 多媒体彩铃逻辑分层设计图

Service 层(业务功能处理单元): 业务功能处理单元是指能够完成相对于用户来说一个完成功能的模块。比如，开户、销户、定制铃音。是从用户的角度去组织该层的结构。是底层向上层(各种接入 Portal)提供服务的接口。也就是通常意义的 Service 层。

Module 层(业务逻辑原子处理层): 为了尽可能的统一业务功能处理单元的的代码，让各个省市的接入 Portal 和业务逻辑处理层最大解耦，提供了业务逻辑原

子处理层。业务逻辑原子处理层存在的必要性是需要屏蔽各省市业务数据的异构性和屏蔽各个省市业务流程的差异性。该层整合了数据层操作的数据形式，并按照一定的形式转化为 Portal 所需要的数据形式并返回给 Service 层。保证 Service 层对 Portal 层提供统一的数据形式，解决了数据差异性。另外该层完成业务功能原子操作，例如获取铃音信息、删除一条铃音设置规则、判断用户是否可以下载铃音和对用户进行扣费操作。该模块类似一块块积木，由业务功能处理单元调用。业务功能处理单元类似就是一块块积木搭起来的一个模块，通过模块化的调用，可以解决流程的异构性。Module 层本身作为单元操作模块，有可能产生很多细小函数模块；但从软件整体考虑，由于其属于内部接口，与 Service 层不同的是，接口的复用性要高于可读性，因此应尽量将功能相同或相似的模块进行合并，保证代码复用。

Dao(数据层): 彩铃 Portal 管理的是彩铃数据库中的数据资源，所以数据层的数据至关重要。数据层封装了对数据库的操作，采用 Spring + Hibernate 框架，事务管理采用 Spring 的可声明式事务管理，通过配置完成对事务的管理。

2.4 异常处理设计

从软件工程开发角度讲，每一个软件单元，小到单一的方法函数，大到提供复杂功能的软件系统都需要有合理清晰的异常设计和处理流程^[7]。所以在多媒体彩铃 Portal 接口开发中，异常子系统的设计就必不可少。最简单的异常设计就是在业务逻辑处理中检查数据和逻辑，如果出现问题就抛出异常，然后由上层进行捕捉和处理。这样的设计方式有很大的缺陷。第一由于异常并没有具体分类，上层不能具体分析异常原因，不能做出对应的处理；第二由于异常的检测和抛出穿插在业务逻辑中，对业务逻辑是一种侵入，会导致代码可复用性差，可读性差，不易维护和变更等。所以我们需要一种分类明确，与业务逻辑分离的异常处理流程。

在 UP 框架中，所有的异常如图 6 所示分为四个大类，分别是代表非法操作，无权限，对象不存在和参数错误。他们都是继承自一个命名为 ServiceException 的抽象异常。每一类的异常再可以扩展各种具体的异常。那么在 Portal 侧只需要捕捉 ServiceException 并判断它的类型就可以明确的判断出异常的原因，并针对异常做出相应处理。

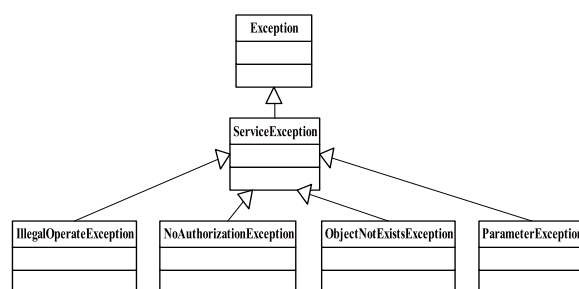


图 6 多媒体彩铃异常结构设计图

那么异常在 UP 中又是如何抛出的呢？由于分层结构中加入了模块化的业务逻辑原子处理层即 Module 层，异常的管理就可以依靠 Module 层的某个模块实现。正如 2.1 节中我们讨论的一样我们将所有可能抛出错误的验证方法集中到一个 RuleModule 的模块中。Service 在进行业务之前，先调用 RuleModule 进行判断。这样所有的异常流程都被封装在一个原子的模块中，跟业务代码完全分离，而且需要增加或修改异常流程对于其他的模块并没有影响，达到了我们异常设计的预期效果。

3 UP 框架的应用和推广

本节将以开发复制好友铃音接口为例，讲述如何基于所设计的 UP 框架进行需求开发，以及这种开发模式的一些特性。多媒体彩铃用户 A 在使用多媒体彩铃业务时，可以通过短信 Portal，Web Portal，IVR Portal 等接入方式查看自己好友 B 的多媒体彩铃资源。A 可以通过复制的方式将 B 的多媒体彩铃加入到自己的多媒体铃音库中来。

3.1 UP 框架的应用流程

基于 UP 架构的开发的步骤可以分以下几步：1) 确定数据结构；2) 定义异常情况；3) 确定业务流程；4) 完成代码开发。

首先复制功能所涉及的两个用户和用户订购信息属于 Portal Date 数据，所以需要 Service 层调用 Module 得到 Portal 层所用的 Portal Date。可能出现的异常有：没有复制权限、用户 A 已经拥有铃音等。这些异常的判断和抛出都可以封装到 RuleModule 的方法中。然后业务的流程应该是：得到两个用户数据，得到目标用户的铃音资源，判断是否可以复制，将铃音资源加入到请求用户的铃音库中，计费。最后代码可以处理成下面的方式。

```
public void copyRing(String phoneNumber, String
```

```

targetNumber, String ringId){
    PortalUser user =
userModule.getUser(phoneNumber);
    PortalUser targetUser =
userModule.getUser(targetNumber);
    PortalDepot targetDepot =
userDepotModule.getDepot(targetUser,ringId);
    Chargeable chargeable =
targetDepot.getChargeable();
    ruleModule.canCopy(user, targetUser,
chargeable);
    userDepotModule.addDepot(user, chargeable,
null);
    chargeModule.chargingForCopy(user,
targetUser, chargeable);
}

```

最后需要完成 Module 的方法,并提供与底层数据交互的 Dao 方法就可以了。从 Portal 侧看,复制的业务功能只需要两个号码和一个铃音资源编号就可以了。Service 的实现过程中则是使用了可复用的 Module 方法,将 Module 模块拼搭出可以实现需求的业务逻辑,其中异常处理完全内聚到一个 Module 模块中。而 Dao 层只需要关心数据库中真实的数据形式,而不必关心与 Portal 侧的数据转换。同样这些 Module 不只是在这个接口中使用。例如订购铃声的需求就可以使用“将铃音资源加入铃音库”的模块方法。而且如果这个接口在其他地区需要做出改动,我们只需要改动模块的调用顺序或添加删除模块就可以实现了,而不用重新开发一整套的业务逻辑。

UP 框架的使用使得接口的开发更方便更快捷,使得接口在使用中可以完全统一。使用 UP 框架开发的接口在四川、浙江等省份的多媒体彩铃项目中已经得到验证,并取得很好的效果。从开发角度,实现了去除数据及流程的耦合性,将功能做到内聚,提供最简单的接口;从维护角度,保证最小的资源浪费和最大程度上的代码重用。

3.2 UP 框架的推广

UP 框架是在多媒体彩铃 Portal 接口开发中提出的,但它具有和其他项目共通的部分。Portal 从广义上讲是和用户交互的门户,只要一个软件或系统需要有其他角色的参与就需要有和其他角色交互的部分,这部分就是 Portal^[8]。同样为了实现存储和动态功能,数据库是不可缺少的部分。Portal 是角色和数据交互的部分,提供给角色一定的功能集合。当今的 Portal 呈现

出多样化,个性化的趋势。每个系统都存在多个交互的 Portal,相同类型的 Portal 在不同的地点、时间可能会出现不同的要求。在这种 Portal 的开发中 UP 开发框架是十分适合的。无论 Portal 数据形式如何不同,他们所使用数据可以通过 UP 框架中的数据抽象统一起来。无论业务功能如何改变,他们可以通过 UP 框架中模块化的原子功能层次构建出来。同时 UP 框架提供了一种内聚性很高的异常处理方式,保证 Portal 系统的健壮性和完整性。这种非侵入的设计和 Spring^[9]的 AOP(Aspect Oriented Programming,面向切面编程)的设计理念异曲同工,可以扩展到日志管理等切面性的操作中,从而扩展 UP 框架的设计方式。

4 结束语

随着社会和科技的进步,人们的交互手段逐渐增多,软件系统的开发也必须在新的环境下有新的发展。本文提出的 UP 框架正是基于这种发展的形式,在原有 Portal 开发框架的基础上形成的新的开发模式。在大型的软件和系统中,软件的可复用性和统一性尤其重要,所以 UP 框架的设计虽然增加了代码的复杂性,但这种设计所带来的优点是显而易见的。UP 开发框架是在真实的项目中抽象出来的开发模式,在对其他类似 Portal 接入系统设计时,可以借鉴本文的 UP 开发框架方案。

参考文献

- 1 沈奇威,廖建新,王纯,蔡斌.多媒体回铃音业务研究.计算机工程,2006,9:231-236.
- 2 隋毅,廖建新,朱晓民,等.彩铃语音搜索功能设计与实现.北京工商大学学报,2007,25(6):35-38.
- 3 王迪明,朱晓民,廖建新,等.彩铃业务 WWW 接入系统的改进.电信工程技术与标准化,2008,4:85-90.
- 4 徐磊,廖建新,王纯.彩铃统一管理业务生成平台的设计与实现.计算机系统应用,2007,16(10):2-6.
- 5 张美琳,王纯,廖建新.彩铃业务管理系统改进.计算机系统应用,2009,18(3):23-26.
- 6 王宇,王力,李增智,王海,班世敏.面向对象软件框架中的耦合性及其消解策略的研究.小型微型计算机系统,2003,10.
- 7 倪奇智,张为华,臧斌宇,朱传琪.保证 Java 精确异常的软件流水线技术.计算机应用与软件,2008,2:21-23.
- 8 王昕.PORTAL (门户)概览.信息技术与信息化,2007,1:31-33.
- 9 Spring Source. About Spring. [2008-05-11]. <http://www.springframework.org/about>