

家校通业务系统结构设计

The system design for communication between home and school

方路平 岑文初 (浙江工业大学软件学院 310000)

摘要:论文通过使用用例驱动的面向对象建模方法,结合 UML 建模工具,实现了家校通业务系统的需求分析、框架结构和流程设计。同时,使用 J2EE 的设计框架和面向对象设计中的成熟模式构建了面向客户的 WEB 平台,使 WEB 平台有良好的处理性能。最后论文通过使用 J2SE 中的 NIO 异步套接字、线程池等提高系统底层数据交互性能,保证了系统在多客户终端的大压力下可以高效运作和响应。

关键词:J2EE J2SE 用例驱动 UML

1 引言

随着社会的进步,当前电脑的不断普及以及 Internet 的广泛运用,网络化信息管理越来越被社会的方方面面所认同,其高效便捷性,无地域限制的特点也成为当前各行各业提高自身管理效率的解决途径之一。与此同时,手机的普及以及短信业务的广泛运用,给人们的生活带来了极大的便利。结合这两种技术运用到教育系统的具体实践中,就产生了“家校通”业务系统。家校通业务系统建立起学校、老师与家庭同步教育孩子的平台,这对科学发展教育、提高教育质量、培养合格人才具有重大的意义。

2 家校通业务系统的功能与结构设计

2.1 系统功能划分

系统总体功能可以划分为四部分:系统信息管理,终端业务处理,短信协议处理,系统上行短信处理。

系统信息管理主要功能是提供了 WEB 平台作为管理员和学校老师的信息维护操作平台,基于系统信息管理可以实现对系统信息的维护,包括了服务提供商管理,系统安全管理,系统日志管理,告警管理;同时也实现了学校信息的维护,包括了学校信息管理,老师信息管理,学生信息管理,信息发布管理,考勤管理,学生在校情况管理,意见反馈,统计分析。

终端业务处理主要功能是提供了对终端机基本指

令操作的支持和在线升级以及下载信息的支持。

短信协议处理用例主要功能是提供了对 SGIP 协议和 CMPP 协议的封装,通过配置可以在系统启动的时候确认当前系统所使用的是哪一种短消息接口协议。

系统上行短信处理用例主要功能是提供了对手机侧短信上发到平台的业务处理。

2.2 系统模块结构设计

系统总体结构见图 1。

系统划分为 4 个功能模块:终端业务处理模块,短信协议处理模块,系统上行短信处理模块,系统信息管理模块。

3 各个模块结构设计

3.1 终端业务处理模块

开通业务的学校会被安置几部家校通终端,家校通终端类似于公用电话,提供了液晶屏和按键,提供用户刷卡和操作。终端机通过 GPRS 无线上网,连接到 Internet 上的系统终端业务处理服务模块,通过 TCP/IP 协议进行数据交互,数据格式是内部定义的消息接口。

终端业务处理模块根据功能可以分成两部分:终端基本命令响应,终端在线内容升级与下载。终端基本命令响应多半是简单的应答模式,服务端会与数据库交互来获取一些用户信息以及终端信息。终端在线内容升级以及下载,是多次交互的数据报传输过程,用

来在线升级终端机内部程序和下载终端机所属学校的老师和学生信息到本地。

在模块结构设计中将底层数据通信分析,数据库基本操作封装,业务逻辑实现分别独立开来,业务逻辑实现基于底层数据通信分析的结果来进行逻辑处理,同时会调用数据库基本操作来进行数据查询和记录。

3.1.1 底层数据通信分析

底层数据通信分析部分的设计一共考虑两套方案:

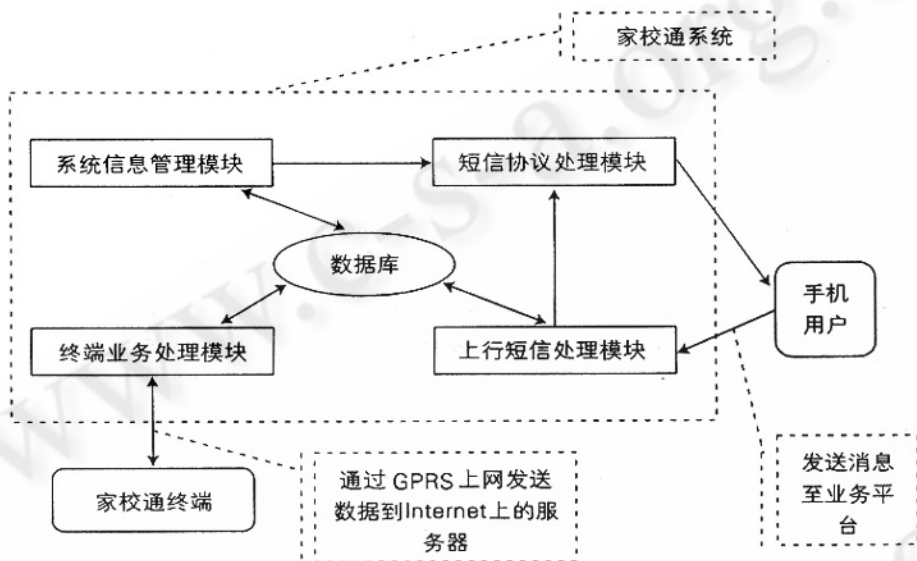


图 1 系统总体功能结构图

(1) 利用多线程客户端处理结构。

当客户端连接到服务器时,服务器就生成一个子线程,并且将子线程注册在系统内,作为以后控制和回收子线程的依据,子线程初始化内部的一套处理资源(接收缓存,发送列表等),配置基本的处理参数(循环检查时间间隔,垃圾数据核准依据时间,接收缓存业务处理最小包长等),然后就开始循环处理接收和发送数据信息。当客户端发送数据到服务端,那么将会被放入到接收缓存,同时检查接收缓存是否已满,做丢弃数据或者等待处理,然后就是上层数据业务分析模块对缓存信息进行处理,处理以后做出回应,将处理结果挂在发送列表上,同时清除已分析的数据。子线程定时循环检查发送列表和接收缓存,如果发现接收列表中

有数据,就发送到客户端,如果发现接收缓存内的数据为长期未处理的垃圾数据,就清除这些数据。当客户端断开时,服务端根据注册信息获取子线程,将线程资源释放,并且结束子线程。

(2) 利用 J2SE1.4 版本提供的 NIO 包中非阻塞服务器结构。

Java NIO 非堵塞技术实际是采取 Reactor 模式,或者说是 Observer 模式为我们监察 I/O 端口,如果有内容进来,会自动通知,这样就不必开启多个线程死等,从外界看,实现了流畅的 I/O 读写,不堵塞了。

比较这两种方式主要的差别就在于前者是理论上并行处理(根据计算机 CPU 的时间片来分配给多个线程达到类似于并行效果),后者是串行处理。在资源分配和管理上前者主要是线程分配自己管理,可以比较灵活,但是出现异常情况容易造成资源无法回收,后者由服务端统一分配统一管理,资源容易管理,但是相对来说不是很灵活。同时,后者对于线程资源的节省也对 CPU 处理提供了

较大提升。

家校通业务模型比较特殊,具有峰谷情景,当学生上学放学的时候,客户端处于连接状态,学生在某一个时间段会集中刷卡,这样使用 NIO 服务器模式可能会造成无法处理而产生拥堵的状况,而并行处理将会缓解这种情况,因此在终端业务处理模块中选用的是第一种服务器处理模式。而在后面的上行短信处理模块中,同样是服务器模块设计,将会采用 NIO 模式,因为上行短信时间比较分散,NIO 可以充分利用系统资源,节约资源。

3.1.2 数据库基本操作

在家校通系统中使用的是 Oracle 数据库,通过 Ja-

va 的 JDBC 连接模式来操作数据库,考虑到性能问题,使用通过上下文获取数据源,然后产生数据库连接池来管理数据库连接,因为有比较成熟的数据库连接池的开源工程,因此就使用了 Apache Jakarta 项目组的 DBCP(Database Connection Pools)。在系统设计中对于 DBCP 包支持的基本操作使用代理类作为封装并且作了部分的功能性增强,预处理和后处理。其作用就是使系统的业务逻辑脱离具体的底层数据操作逻辑,达到较好的松耦合。

系统中首先定义了数据库操作接口 (IDBManagement),接口中定义了基本通用的数据库操作例如 ExecSqlUpdate, ExecSqlQuery, ExecSqlBatch, ExecProc, ExecFunc 等,依次为执行数据库更新操作,执行数据库查询操作,执行数据库批量操作,执行数据库存储过程,执行数据库函数等。系统数据库操作定义类中包含了 JDBC 标准数据库操作实现的一组对象例如 connection(数据库连接),resultset(数据库结果集),Prepared-Statement(预处理数据库操作执行对象)等,实现了数据库操作接口定义(通过使用 JDBC 提供的基本数据库操作函数来实现),同时提供不同参数和对返回的结果集的预处理和后处理,方便系统调用。

3.1.3 业务逻辑实现

由于家校通终端设备由第三方提供,终端通过 GPRS 无线上网与平台通过 TCP/IP 协议进行数据流交互,平台必须根据终端生产厂家给出的终端操作命令接口协议来实现终端各种指令的响应以及业务流程的处理。该模块的业务逻辑主要就是要实现对终端简单操作命令的响应以及对终端多次交互流程的支持。因此首先根据终端厂家提供的接口设计了系统的外部业务逻辑处理接口。然后定义了一个工厂类,作为终端接口生产类。因为考虑到可能以后会支持多种不同的终端,业务处理逻辑可能不同,那么只要支持统一的接口函数,通过工厂类在选择生成的具体终端处理实现类,通过面向对象的多态性,就不需要修改服务端的代码直接可以插入一种新的终端类型来进行处理。参见图 2。

其中产品类就是业务逻辑的具体实现类,产品接口就是业务逻辑接口,工厂类就是产生产品接口的生产者。

3.2 短信协议处理模块

短信协议处理模块主要功能就是协议分析与封装,提供给系统内部或者第三方协议数据处理接口,方便的接收和发送短消息。主要功能模块也就分为三部分,基本参数和配置的载入,协议分析和封装,外部系统接口调用(可以提供 Webservice 的远程接口调用)。

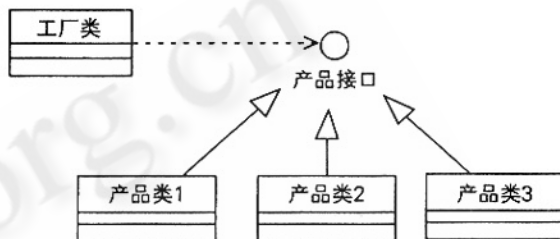


图 2 业务逻辑多态性的运用

该模块的重点就是分析 SGIP (Short Message Gateway Interface Protocol) 协议和 CMPP (China Mobile Peer to Peer) 协议的具体内容和流程来定义协议处理接口和实现协议处理逻辑。然后基于协议封装接口提供用户接口。

3.3 系统上行短信处理模块

系统上行短信处理模块设计十分类似于终端业务处理模块,同样在模块结构设计中将底层数据通信分析,数据库基本操作封装,业务逻辑实现分别独立开来。其中底层数据通信分析部分使用的是在终端业务处理模块设计中提到的 NIO 非阻塞服务器结构。数据库基本操作封装可以共用终端业务处理模块的部分。业务逻辑设计也是首先定义处理逻辑接口并作具体的实现类来实现业务逻辑内部流程。

系统将上行短信集中放置在系统开辟的短信池,连接上来的客户端必须在系统注册,同时提供自己的业务类型标示,当短信池里面相应业务类型标示的短信存在时,系统就会主动通知客户端收取短信,并且做出处理。接收池结构图见图 3。

首先系统定义了短消息接收池接口 (ISMPool) 定义了注册,注销,通知观察者三个操作。短消息接收池实现类除了实现接口定义的操作以外还提供了查询当前所有注册的观察者列表。定义了观察者接口,接口定义

了自我更新处理函数,当被观察者发生变化时,它会根据注册的观察者来选择通知观察者调用自我更新处理,也就是说,当有上行短消息发送到短消息接收池中,短消息接收池会选择注册的某部分客户端,将短消息内容作为参数提交给客户端,并唤醒客户端进行处理。

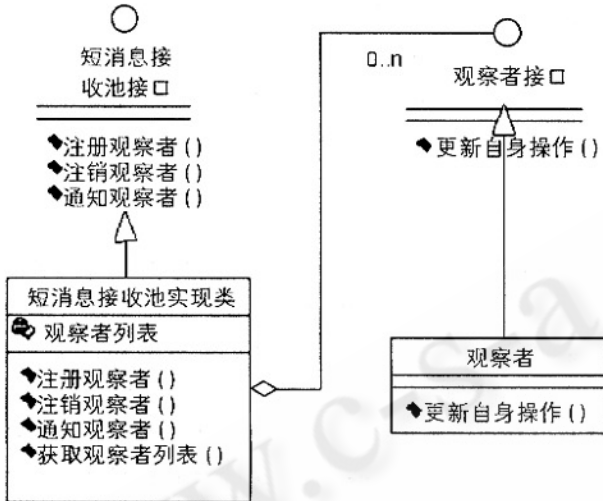


图 3 上行短信处理模块中短信接收池架构图

在上面描述的结构中,主要的两个角色就是观察者和被观察者,其中分别被抽象成为了短消息接受池接口和观察者接口。被观察者主要的需要实现的接口操作就是对观察者列表的一个维护,包括了注销,注册。同时,当被观察者内部资源发生变化的时候可以通知特定的观察者,通知主要依靠观察者自身所实现的接口操作定义中自我更新操作。而观察者需要实现的就只是一个当被通知以后如何作出处理的自我更新操作。因此,所谓的观察,其实是通过被观察的反向通知来实现观察,而不是通过正向查询来获得被观察者状态的改变信息,这样的倒置通知有利于简化资源状态管理。

3.4 系统信息管理模块

系统信息管理模块分成两大部分,管理员操作模块和业务信息管理模块,前者主要是系统管理员对系统信息和资料的管理,后者提供了管理和构建学校,老师,学生信息,管理维护学生考勤以及平台信息发布等功能。

系统信息管理模块主要的用户界面就是 WEB 方式,因此系统信息管理模块就使用了 Struts 框架模型来构建 Web 平台,采取 MVC (Model, View, Controller) 的结构体系。

视图部分 (View) 就是一系列提交到客户浏览器进行显示的 JSP 文件,是用户看到并与之交互的界面。在这些界面中没有业务逻辑,也没有模型信息,只有标签。

模型部分 (Model) 表示应用程序的状态和业务逻辑,定义了业务逻辑接口和业务逻辑实现部分。在业务逻辑处理中也会调用数据库基本操作接口。

控制器 (Controller) 的作用是从客户端接受请求,并且选择执行相应的业务逻辑,然后把响应结果送回客户端。在 Struts 中 Controller 由 ActionServlet 类和 Action 类来实现。ActionServlet 主要负责接受 HTTP 请求信息,Action 类负责调用模型的方法,更新模型的状态,并帮助控制应用程序的流程。

除了使用框架结构作为系统的总体系统架构以外,还是用了 Struts 的 Tile 用来定制标准化的一些页面结构,增加了页面的重用性和可扩展性。

4 总结

在家校通业务系统设计中使用 J2EE 多层结构模式来构建 Web 信息管理平台,使 Web 系统结构灵活可扩展,使用多种面向对象编程模式,并且运用到系统开发过程中,提高了程序的灵活性和可扩展可维护性。

参考文献

- 1 中国移动通信公司,中国移动通信互联网短信网关接口协议(CMPP v2.0)[S],2002。
- 2 中国联通通信公司,中国联通短消息网关联网协议(SGIP V1.2)[S],2003。
- 3 阎宏,Java 与模式[M],北京 电子工业出版社,2002。
- 4 Ron Hitchens,Java NIO[M],O'Reilly&Associates Inc,2002。
- 5 Bruce Eckel,Java 编程思想 第3版[M],陈昊鹏、饶若楠译,北京 机械工业出版社,2005。