

基于 B/S 架构的呼叫中心业务系统框架设计

林海, 武波

(西安电子科技大学 软件工程研究所, 陕西 西安 710071)

摘要:针对因业务需求的多样性导致的呼叫中心业务系统开发中存在的诸如重复开发的问题,提出了一种通用的呼叫中心业务系统框架的设计方案。本方案对各种业务系统进行了抽象,形成了通用的系统管理框架。该框架与业务无关,具有很强的通用性;并且该框架基于 B/S 的架构,易于扩展,满足了需求不断变化的呼叫中心业务系统的要求。开发者可以在该系统框架的基础上,开发与具体业务逻辑相关的功能模块,形成一个完整的呼叫中心业务系统。

关键词:呼叫中心;业务系统;B/S 架构;MVC 模式;系统框架设计

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2008)08-0202-04

Framework Design of Call Center Business System Based on B/S Structure

LIN Hai, WU Bo

(Software Engineering Institute, Xidian University, Xi'an 710071, China)

Abstract: A design scheme of universal call center business system framework, aimed at the problems exist in development of call center business system such as repeat development caused by variety of business requirement, is put forward. The scheme abstracts various business systems, and forms a universal system management framework. Having no relationship with business, the system framework has strong versatility, and being based on B/S structure, the system framework can be easily extended and meets the call center business system's demand of changing requirement constantly. Based on the business system framework, developers can develop functional models concerned with idiographic business logic, and then form an integrated call center business system.

Key words: call center; business system; B/S structure; MVC pattern; design of system framework

0 引言

呼叫中心也称客户服务中心,是一种将计算机和电话集成技术、交互式语音响应技术同计算机信息处理系统相结合的一种应用,它提供了一种新型的信息服务手段^[1]。呼叫中心应用于广泛的领域,例如航空服务呼叫中心、114 查询中心、1001 联通客服热线、1860 移动客服热线等都应用到了呼叫中心系统^[2]。呼叫中心业务系统是指在独立于不同行业应用逻辑的话路支撑系统上建立的针对具体行业应用逻辑的业务处理系统^[1]。

传统的呼叫中心业务系统一般采用的是 C/S 架构。C/S 架构的缺点在于:(1)需要开发 Server 和 Client 两端,开发复杂度高;(2)开发采用的协议不同,

不利于维护;(3)可扩充性差,需要增加新的功能时工作量大。

基于 B/S 架构的系统正可以弥补 C/S 架构的以上缺点,B/S 架构的优点是:(1)B/S 架构只需开发 Server 端程序;(2)B/S 架构均采用统一的 TCP/IP 协议,便于维护;(3)B/S 架构可扩充性强,完全满足呼叫中心业务系统的需求^[3]。

通过以上的叙述和对比,不难看出呼叫中心业务系统的开发具有很强的现实意义,而 B/S 架构的优点使其成为设计呼叫中心业务系统的不二选择。

1 呼叫中心业务系统框架设计可行性分析

行业不同,它们的具体业务也必然存在着的差异,那么设计一个通用的呼叫中心业务系统框架有没有可能呢?纵观各个行业呼叫中心的业务流程(如图 1 所示),抛开具体的业务应用,可以看到它们的业务流程是相似的。

不仅业务流程相似,不同行业呼叫中心的业务系

收稿日期:2007-11-09

基金项目:国家自然科学基金(60573034)

作者简介:林海(1982-),男,山东临沂人,硕士研究生,研究方向为互联网技术;武波,教授,硕士生导师,研究方向为软件工程和软件设计理论与应用。

统除了具体的业务模块不同以外,剩下的部分,例如系统管理、功能管理、用户管理、角色管理等,设计和实现方式虽然有可能存在着差异,但其提供的功能都是大同小异。

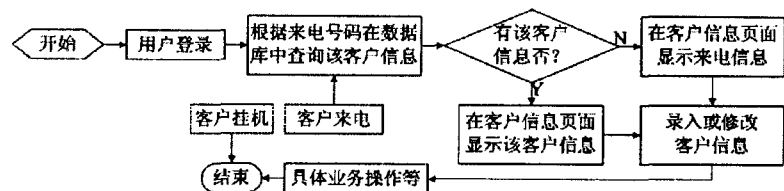


图1 呼叫中心业务流程

从以上两个相似性可以看出,设计一套通用的呼叫中心业务系统框架是可能的。该业务系统框架不包括具体的业务模块,只着眼于各业务系统必需的功能管理、用户管理、角色管理等系统管理模块。开发者可以在此系统框架的基础上扩展业务应用模块,形成一个完整的呼叫中心业务系统。

2 呼叫中心业务系统框架的设计

该呼叫中心业务系统框架采用MVC模式进行设计,MVC模式的优点在于它不仅实现了功能模块和显示模块的分离,同时它还提高了应用系统的可维护性、可扩展性、可移植性和组件的可复用性。图2所示为该业务系统框架的各个模块和层次。

从功能上,该框架有功能管理、子功能管理、角色管理、用户管理、授权管理、角色分配管理等共六个管理模块;从MVC的层次划分上,各个模块又由属于不同层次的文件组成。在该框架的六个模块中:

(1)功能管理模块负责管理业务系统框架提供的各个功能模块,每个功能模块又包含多个子功能;(2)子功能管理模块负责管理业务系统框架提供的各个具体的子功能项,每个子功能项隶属于一个功能模块;(3)角色管理模块负责管理访问业务系统框架的角色,不同的角色被授予不同的权限,拥有某项权限的角色可以访问与该权限对应的子功能项,对访问系统框架的各个角色的授权由授权管理模

块负责管理;(4)用户管理模块负责管理访问业务系统框架的各用户的信息,不同的用户被分配给若干个角色,每个用户所具有的权限是他被分配的所有角色对应的权限集合的并集,对访问系统的用户分配不同的

角色由角色分配管理模块负责管理。

下面对各模块进行详细阐述。

功能管理模块负责管理业务系统框架的各个模块,图3所示为增加和删除一个功能模块的序列图。

在功能管理模块中,增加一个新的功能模块比较简单,但是要注意:增加一个新的模块后,新模块并不包含任何子功能项,只是一个“空架子”而已;在删除一个功能模块时,不仅该功能模块的信息,该功能模块包含的子功能项的集合,以及与该集合中各子功能项对应的授权信息都要删除。

增加了新的功能后,就要给它增加子功能项,这由子功能管理模块负责,子功能管理模块还负责更新已有的子功能项,如图4所示。

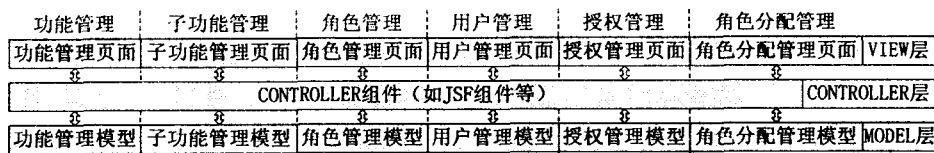


图2 系统框架模块层次示意图

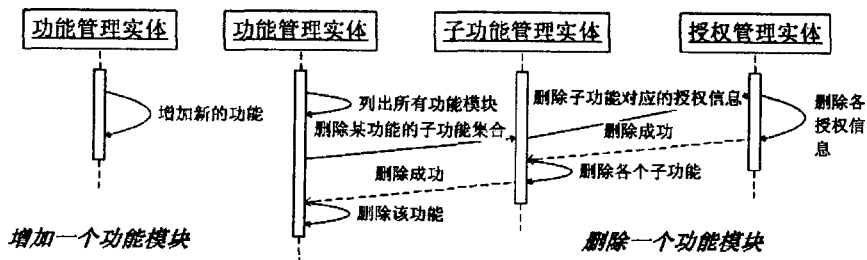


图3 增加和删除功能模块的序列图

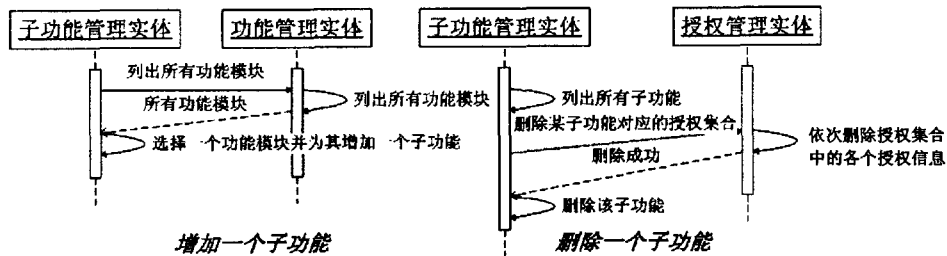


图4 增加和删除子功能的序列图

在增加子功能时,要先选择功能模块,因为每个子功能都隶属于某个功能模块;同删除功能模块一样,删除子功能时,也要把它关联的授权信息删除掉。

在本设计方案中,使用业务系统的用户是分角色的,不同的角色通过授权可以访问不同的子功能。对角色的管理是由角色管理模块负责的。

增加角色的流程和增加功能模块(见图 3)的几乎一样,因此在图 5 中仅示意出删除一个角色的序列图。

在删除一个角色时,与该角色有关的授权信息(“角色—子功能”对)以及角色分配信息(“角色—用户”对)也会被关联删除。

“角色—子功能”对,也即授权信息是由授权管理模块负责管理的。

图 6 所示为增加和删除某一授权信息的序列图,因为一个授权信息对应着一个角色和一项子功能,所以在增加一个授权时要列出所有的角色和所有的子功能。值得注意的是,如图 6 中所示,授权管理模块在删除某一个授权时有两种依据,一种是依据子功能,一种是依据角色列出授权信息的集合,在该集合中选择一个授权删除之。

访问业务系统框架的用户信息由用户管理模块负责管理,因为增加用户的流程与增加功能模块和增加角色的流程非常类似,而删除用户的流程同删除子功能项(见图 4)的流程非常类似,只是把图中的“子功能”改为“用户”,把“授权”改为“角色分配”即可。因此增加和删除用户的序列图不再给出。

角色分配管理模块负责给访问系统的用户分配不同的角色。给一个用户多分配一个角色的流程和增加一个授权信息的流程很类似;而删除分配给某用户的一个角色的流程同删除一个授权信息的流程也很类似,因此只要在图 6 中把“授权”改为“角色分配”、把“子功能”改为“用户”就可以看到增加和删除角色分配的序列图。在删除角色分配时也可以选择依据角色还是依据用户列出角色分配的集合。

2.1 Model 层数据模型的设计

根据以上的系统模块层次示意图和各模块实现其功能的序列图,可以设计出六个管理模块的实体模型。

功能管理实体模型由类 Functions 描述,子功能管理实体模型由类 FunctionItems 描述,角色管理实体模型由类 Roles 描述,用户管理实体模型由类 Users 描述,授权管理实体模型由类 Authorities 描述,角色分配

管理实体模型由类 Userroles 描述。六个类的数据成员和成员函数以及类之间的关系如图 7 所示。

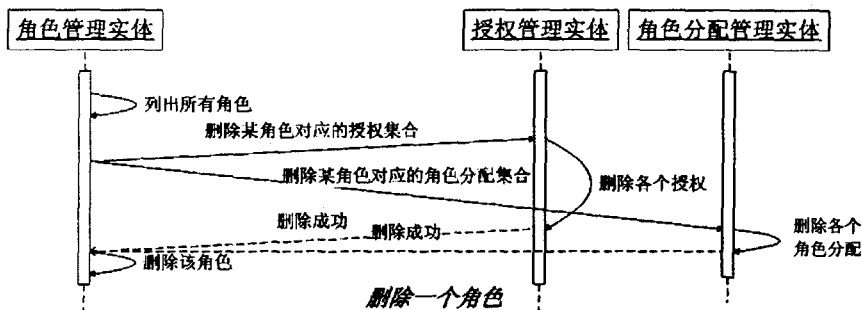


图 5 删除角色的序列图

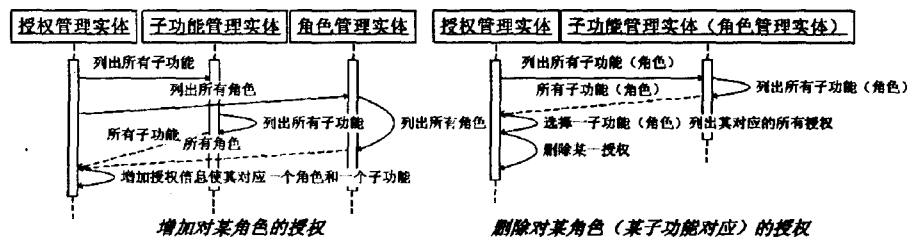


图 6 增加和删除授权的序列图

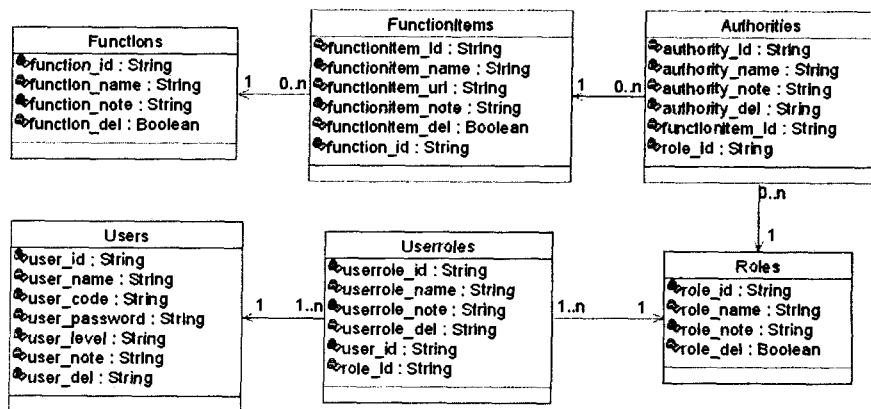


图 7 基本类与类之间的关系

每个功能模块包含若干子功能项,因此 Functions 类与 FunctionItems 类是 1 对 0 到多的关系。每个角色可以被赋予多个权限(也即子功能项),每个权限也可以被赋予多个角色,因此角色与子功能项之间是多对多的关系。

多对多关系一般通过增加一个中间表分解为两个一对多关系,因此 Roles 类与 FunctionItems 类的多对多关系通过 Authorities 类分解为两个一对多关系: Roles 类和 Authorities 类的 1 对 0 到多的关系; FunctionItems 类和 Authorities 类的 1 对 0 到多的关系。

同理, Users 类和 Roles 类的多对多关系通过类 Userroles 分解为两个一对多关系: Users 类和 Userroles 类的 1 对 0 到多关系以及类 Roles 和类 Userroles 的 1 对 0 到多关系。

2.2 实体模型对应的数据库表结构的设计

以上详细介绍了六个管理模块具有的功能以及处于各个模块核心的实体模型的设计,下面简单介绍一下数据库的表结构设计,因为最终系统框架要依靠数据库中的数据才能正常运行。

与六个实体模型对应的六个数据库表结构的定义如下:

(1)功能模块信息表(T_FUNCTIONS):对应 Functions 类,维护该呼叫中心业务系统框架提供的功能模块的信息;

(2)子功能信息表(T_FUNCTIONITEMS):对应 FunctionItems 类,维护该呼叫中心业务系统提供的详细的子功能项的信息,每一个子功能项从属于功能信息表(T_FUNCTIONS)维护的功能模块,每一个子功能项在最终实现的业务系统中对应一个 Web 页面,通过角色授权表(T_AUTHORITIES)维护的信息决定各个角色是否有访问的权力;

(3)角色表(T_ROLES):对应 Roles 类,维护使用该呼叫中心业务系统的角色信息,不同角色被赋予不同权限,这些“角色—权限”对由角色授权表(T_AUTHORITIES)维护;

(4)角色授权表(T_AUTHORITIES):对应 Authorities 类,维护使用该呼叫中心业务系统的“角色—权限”对,这些权限即子功能信息表(T_FUNCTIONITEMS)维护的子功能项;

(5)用户信息表(T_USERS):对应 Users 类,维护使用该呼叫中心业务系统的用户的信息,可以分配给每个用户若干由角色表(T_ROLES)维护的角色信息;

(6)用户角色表(T_USERROLES):对应 Userroles 类,维护该呼叫中心业务系统的“用户—角色”对。

根据六个实体模型之间的关系和图3~图6所示的序列图得到六个表结构之间的关联关系如图8所示。

如图8所示,每个子功能项(表 T_FUNCTIONITEMS 维护)都属于一个功能模块(表 T_FUNCTIONS 维护),因此表 T_FUNCTIONS 的主键 function_id 作为表 T_FUNCTIONITEMS 的外键;

每个授权项(表 T_AUTHORITIES 维护)都与一个子功能项和一个角色(表 T_ROLES 维护)对应,因此表 T_FUNCTIONITEMS 的主键 functionitem_id 和表 T_ROLES 的主键 role_id 作为表 T_AUTHORITIES 的两个外键;

每个角色分配项(表 T_USERROLES 维护)都与

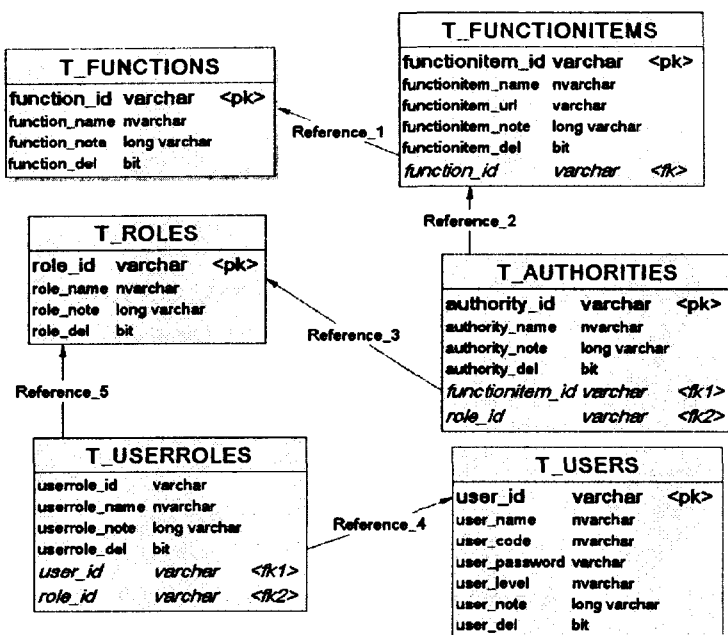


图8 表结构和表间关系

一个角色和一个用户(表 T_USERS 维护)对应,因此表 T_ROLES 的主键 role_id 和表 T_USERS 的主键 user_id 作为表 T_USERROLES 的两个外键。

2.3 View 层和 Controller 层的设计

Web 应用中的 View 层是由多个 Web 页面组成的。该呼叫中心业务系统框架提供的每个管理模块都对应两个 Web 页面:信息录入页面和信息维护页面。信息录入页面负责其对应的管理模块中增加记录的功能;信息维护页面负责其对应的管理模块中查询、更新、删除记录的功能。

Controller 层可以理解为从用户接收请求,将 Model 层的模型与 View 层的视图匹配在一起,共同完成用户的请求^[4]。它就是一个分发器,选择什么样的模型,选择什么样的视图,可以完成什么样的用户请求。控制层并不做任何的数据处理^[5]。因此,在本设计方案中,系统框架的 Controller 层的设计比较简单,利用现成的组件来构成。比如采用 JavaServer Faces 技术,View 层的各个 Web 页面和 Model 层的各个实体模型类只需要简单地注册到 JSF 的 XML 配置文件中,JSF 组件就可以将 View 层的 Web 页面(视图)和 Model 层的实体模型匹配,完成 Controller 层的功能。

以上分析了呼叫中心业务系统框架的设计思路,组成该业务系统框架的六个管理模块要实现的功能以及六个实体模型对应的数据库表结构,但并未涉及该系统框架中的实体模型是如何操作数据库的。对数据库的操作可以采用各种数据持久化技术把对数据库的操作和业务系统分离,这已超出了文中的范围,在此

(下转第 210 页)

框架下的概念模型,提供规范的标准,以及如何模块化一个 OSGI 的兼容系统,让每一个个性化的服务满足每个用户需求。

应用工程在领域分析的基础上进行实例化,在实例化的过程中有些组件从现有组件库中选取,有些则重新构建,这些组件都是 COM 组件,根据每个系统的具体需求而变更组件,最终实现大粒度的复用。

6 结束语

自从软件危机的提出,人们一直都在寻找高效的软件开发方法,软件复用技术被大家认为是降低成本、缩短开发周期的有效方法。现在的复用技术有基于组件复用技术和产品线开发复用技术等。文中从产品线理论进行研究,对体系结构、开发过程等内容进行了深入的探讨。最后用 HIS 系统举例,提出在 OSGI 框架结构下的应用开发。完善的产品线是一个长期的工作,国外已经有较成熟的产品线,所以我们有责任把产品线理论在国内的软件产业中推广起来。

(上接第 201 页)

5 结束语

变电站电压无功控制(VQC)系统是保障电能质量提高的有效途径,采用优化的九区图控制策略来设计 VQC 系统,弥补了某些区域对控制结果产生振荡现象以及装置频繁动作的缺陷,明显减少了变压器分接头调节和并联补偿电容器组投切次数,增强了电压稳定性和无功补偿的有效性,提高了电能质量,降低了电网损耗,大大改善了电能的利用率。

参考文献:

- [1] 杨 益,方潜生.基于 FPGA 动态跟踪型功率因数补偿控

(上接第 205 页)

不再赘述。

3 结束语

文中探讨了通用的呼叫中心业务系统框架设计的可行性,并提出了一套设计方案。该设计方案采用 MVC 设计模式、基于 B/S 架构,与具体的业务无关,可应用于多种行业,呼叫中心业务系统的开发人员可以在本设计方案的业务系统框架的基础上扩展出与具体业务相关的功能模块。因为可以重复使用、不涉及业务,该设计方案可以减少开发成本,缩短开发周期。根据本设计方案实现的陕西移动 114 号码百事通项目正在实施中,验证了本方案的可行性。

参考文献:

- [1] Mili H. Reuse - Based Software Engineering[M]. 北京:电子工业出版社,2004.
- [2] Bass L, Clements P, Kazman R. Software Architecture in Practice[M]. [s. l.]: Addison Wesley, 1997.
- [3] Mili H, Mili A, Yacoub S, et al. 基于重用的软件工程:技术、组织和控制[M]. 韩 柯译. 北京:电子工业出版社, 2004.
- [4] 杨美清,朱 冰,梅 宏. 软件复用[J]. 软件学报,1995,6 (9):525 - 533.
- [5] Frakes W B, Kang K C. Software reuse research: Status and future[J]. IEEE Transactions on Software Engineering, 2005, 31(7):529 - 536.
- [6] Gornaa H. Designing Software Product Lines with UML: From Use Cases to Pattern - Based Software Architectures [M]. Boston: Addison - Wesley, 2004.
- [7] Chastek G, Donohoe P, McGregor J D. Product line production planning for the home integration system example[M]// Product Line Practice Initiative. [s. l.]: Carnegie Mellon University, 2002.

制器的设计[J]. 工业控制计算机, 2006, 11: 79 - 80.

- [2] 唐 艳,王万新,王洪博,等. 九区图法优化实时无功补偿及其应用[J]. 机械工程与自动化, 2006(4): 154 - 156.
- [3] 康明才. 基于遗传算法的变电站电压 - 无功综合控制[J]. 南京理工大学学报, 2002, 26(5): 490 - 493.
- [4] Celoxica. Handel - C Language Reference Manual[S]. [s. l.]: Celoxica, 2003.
- [5] Celoxica. DK Design Suite User Manual[S]. [s. l.]: Celoxica, 2003.
- [6] 杨 益,方潜生. 基于 Handel - C 的伪随机数发生器的设计与实现[J]. 计算机技术与发展, 2006, 16(12): 124 - 126.

参考文献:

- [1] 耿 刚,陈 平. XML 在呼叫中心业务系统中的应用与研究[J]. 现代电子技术, 2003, 26(24): 31 - 33.
- [2] 许 红. 分布式呼叫中心系统的设计[J]. 甘肃科技, 2006, 22(4): 36 - 38.
- [3] Bernett H, Jaramillo M L. Assessing Web - enabled call center technologies[J]. IT Professional, 2001, 3(3): 24 - 30.
- [4] Lam W, Shankararaman V. An enterprise integration methodology[J]. IT Professional, 2004, 6(2): 40 - 48.
- [5] Cheng I, Boyette N, Krishna V. Towards a Low - Cost High - Quality Service Call Architecture[C]// Services Computing, 2006. SCC '06. IEEE International Conference. [s. l.]: [s. n.], 2006: 261 - 264.