第32卷 第23期 计 算 机 工 程 2006年12月
Vol.32 No.23 Computer Engineering December 2006
·开发研究与设计技术· 文章编号：1000—3428(2006)23—0256—03 文献标识码：A 中图分类号：TP273+.5

商业银行信贷管理系统的设计与实现

梅登华，闵华清
（华南理工大学计算机科学与工程学院，广州 510640）

摘 要：多数商业银行现行信贷系统将业务和流程控制紧密耦合，没有进行很好的分层隔离，在需要对业务流程进行调整的时候就会对程序代码硬性改动，影响系统的稳定性。由于二者的紧密结合，导致整个应用系统无法实施，阻碍了业务发展。各地区之间的系统差异，使得相互之间的业务交互处理不容易兼容。系统不灵活，对一些复杂流程的处理也存在一定的困难。该文通过对商业银行信贷业务的分析，较好地实现了业务和流程控制的分离，并将其成功运用于某商业银行信贷管理系统中。

关键词：信贷管理系统；Struts；Oracle 9i；XML

Design and Implementation of Commercial Bank Credit Management System

MEI Denghua, MIN Huaqing
(Computer Science and Engineering College, South China University of Technology, Guangzhou 510640)

【Abstract】 Business and flow controlling couple with each other tightly in most of commercial bank credit management system. The system is not stable when adjusting the business flow for program code changing. And it is not flexible to baffle the business expansion. It is also difficult to process the complex data flow. The paper analyzes the service of commercial bank credit, separates the service and flow controlling. It is practice effective for a commercial bank credit management system.

【Key words】 Credit management system; Struts; Oracle 9i; XML

信贷管理系统的主要功能实现客户的管理、业务电子化管理、信息交流、统计分析、监测、审批、控制和信息存储、汇总、收集、提供等功能，为各层次的经营管理提供全面的信息及监控、决策工具。系统所搭建的客户信息平台、业务平台，决策支持平台能够成为信息增值(如全面的风险管理)、业务拓展(如推出新的信用组合产品)的基础。为加快信贷业务创新、强化绩效考核手段、降低风险管理成本、提高风险决策效率和水平提供充分的信息支持。

1 系统功能

系统共分为5个模块，分别是客户管理、业务管理、不良资产管理、监控与分析、系统管理。各个模块彼此独立，通过接口实现数据交换。信贷管理系统的主要业务工作为：实现信贷审批电子化，各机构信贷员把客户贷款申请登记后发送到上级主管信贷业务人员，业务人员对这项贷款申请展开调查、审查并填写调查表、审查表，然后提交上级主管信贷领导，领导根据申请中的客户资料、调查情况、审查情况，决定是否批准客户的贷款申请、签署审批意见，然后再发回下级贷款机构，以便下级机构执行，从而实现了贷款审批的电子化。图1为业务管理流程图。系统架构在基于Struts的Castor技术的多层分布式结构基础之上。根据系统的特点，将其分3个层次：表示层，业务逻辑层和数据服务层。在业务逻辑层，根据系统模块设计jdom、jxl、commons-logging、jspsmart、jfreechart等相对稳定的Java组件；针对公共数据访问提供一个公共数据服务comm Java组件；针对数据库操作采用castor组件。当功能升级时，可以通过修改和完善组件用以实现更高级的功能，这些修改可以直接在业务逻辑层进行，既方便用户操作，又能够减少系统维护的工作量。

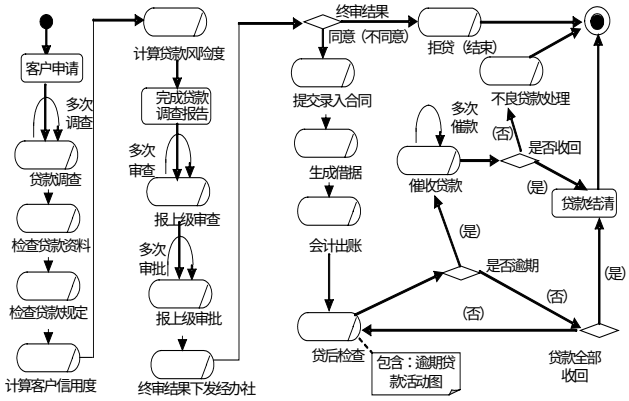

图1 贷款业务流程

图2是信贷管理系统的结构图。

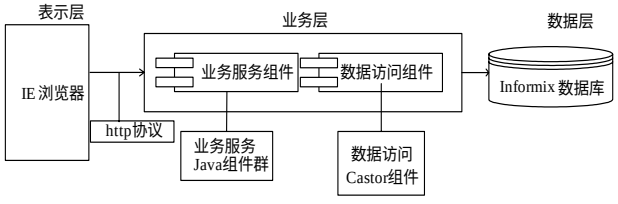

图2 信贷管理系统的结构

在业务层的每个组件可以对应一个或多个业务对象，业务对象在系统中往往映射为一个实体，如员工、角色；也可

作者简介：梅登华(1967－)，男，副教授、博士后，主研方向：软件可靠性，基于Web的软件系统开发，CDMA无线通信系统，嵌入式系统，电力变压器在线控制系统；闵华清，教授

收稿日期：2005-12-23 **E-mail：**dhmei@scyt.edu.cn

能映射为一个关系，如员工和角色的对应；也可能是一个控制对象，如一些复杂的处理，它通过操作其他组件协作以实现某个特定的功能。

2 系统设计

2.1 数据层设计

数据层的主要功能是存储数据。在本系统中分 Oracle 数据和 XML 数据，XML 主要保存贷款调查、审查表的相关数据，其它均保存在 Oracle 内。

信贷管理系统中，数据库中大约有 85 张表。随着系统的运行，数据量逐渐增大，将会影响系统的使用效率和灵活性。在设计数据库程中，针对一个贷款申请，每一份调查和审查报告均保存成一个 XML 文件。

数据层只存放数据本身以及它们的更新、插入和删除等基本操作，不在数据库中存放外键的关联信息，不允许使用存储过程和触发器，与业务相关的所有逻辑均放在业务逻辑层实现。这是面向对象思想和基于组件的思想在整个系统的一个体现，这将保证对每一种数据的操作是完全可控的，因为针对数据库中每一个表的操作必须通过业务层的对应 Java 组件来实现，当系统运行过程中产生异常时，只要修改和更新业务层对应的组件，既可有效控制这种错误的再次发生。

采用存储过程或者触发器将分散应用服务器的业务逻辑功能，导致一个对象的业务逻辑可能通过多种途径实现，增加了维护的难度。采用数据的外键关联虽然可以有效控制数据库的完整性。但是这种关联的后果是：用户在进行操作时，会出现一些莫名其妙的错误提示。将这种外键关联通过业务层实现，可以为用户提供详细的错误信息，指导用户完成业务操作。

2.2 应用服务器结构设计

表现层(客户端)通过业务层接口发出取数命令，业务层接口找到对应的业务逻辑对象，业务逻辑对象向数据库发出取数命令，获取数据，然后再传回给表现层。

考虑到针对数据库的操作就是更新、插入和查询操作，本系统将数据库操作逻辑从业务层独立出来，采用一个共有的数据处理组件 Castor，这个组件命名为数据访问层。另外系统的表现层也不需要知道业务逻辑的实现方式，只需要给表现层提供调用接口，表现层设置对应的参数既可实现相应操作。因此把这个接口独立出来，专门用作表现层访问业务逻辑层的接口，这个接口称为业务表现层。但独立出来的这些部分是逻辑的抽象，在逻辑上它们是多层的实现，在物理上它们仍有可能存在于同一台电脑当中。图 3 是应用服务器结构示意图。

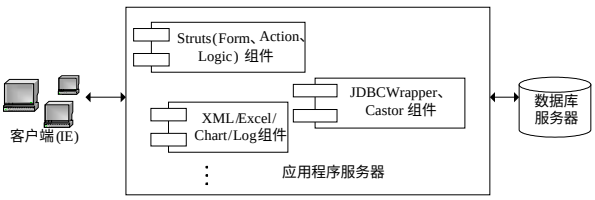

图 3 应用服务器结构

(1)数据访问层

数据访问层为业务服务组件中的实体对象提供各种数据库操作。在本系统中的数据访问组件有 2 类 Castor 和 JDBCWrapper.Class。直接对表的操作时采用 Castor，而在一些简单的更新、统计、查询则采用 JDBCWrapper。

例如在 JDBCWrapper 组件用 Java 创建的数据访问组件中的静态方法为

public static int[] executeUpdate(String[] sql, boolean transRequired) throws SQLException

参数：sql[]数据库查询脚本数组；transRequired 支持事务。

返回：返回执行语句影响的记录元组数，在此基础上按照用户给定的执行语句顺序组成 int[]数组，返回。

用途：提交需要联机事务控制的数据改变，保存变更数据到数据库；典型的为 INSERT、UPDATE、DELETE 等数据库的 SQL DDL 语句。

(2)业务逻辑层

在业务逻辑层中，业务服务组件是以业务划分的，通常是一个业务对应一个业务服务 Java 组件。图 4 是业务逻辑层的构成示意图。其中，Action 控制客户端的请求及转发，Form 映射客户端表单的请求数据项，Data 映射数据库表对象，Logic 负责处理逻辑业务。

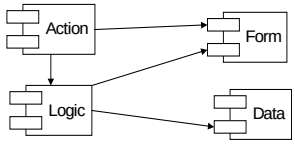

图 4 业务逻辑层的构成

(3)业务表现层

业务表现层的实现比较简单，取消了所谓的对象的概念，只提供了一组封装了业务组件的 JSP 标签。这样实现的好处是：客户端开发人员不需要知道业务组件的定义方式，这便于开发团队分工合作，简化了操作流程，开发人员在使用业务组件前不需要创建业务逻辑组件，只需要调用业务表现层的单元即可。

2.3 客户端界面架构设计

表示层即是客户端界面，是与用户交互的窗口，是系统功能的最终体现。在本系统中，为了规范界面风格，保证系统质量的稳定性和扩展性，提高开发速度，减少客户端维护工作，采用了面向对象的分析方法，提出了统一的界面架构，如图 5 所示。在这个框架下，每个子类窗口由一个到多个 Form 组成，每个 Form 又由 Struts 的 JSP 标签完成相应的任务需求。

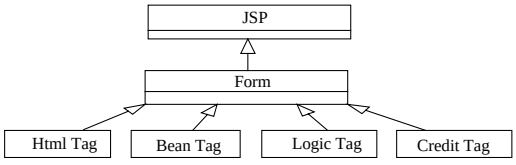

图 5 界面框架的关系

3 系统实现

3.1 业务层的实现

在信贷管理系统中，数据库只负责数据的保存。系统重点放在应用服务器向后台数据库存取数据和应用服务器业务逻辑的实现上。

后台存取数据采用 JDBCWrapper 组件和开源的 Castor 组件来实现，二者互相独立。所有的业务逻辑组件均通过这 2 个数据访问组件，实现与数据库的交互。在实现上，通过 Borland JBuilderX 的工具实现 Web 的开发。

(1)数据访问组件的实现

数据访问组件需要与数据库交互，在实际实现中数据访

问组件在 Castor 中实现，以 Castor JDO 和 Castor XML 建设这个数据访问层：1)通过 mapping.xml 定义数据库到数据对象的映射模型；2)通过 Castor XML 读取 mapping 内的配置文件，将数据表映射到对应的数据对象中；3)用 Castor JDO 对数据对象的操作，反向持久化到数据库。这样，就架构了数据访问组件的实现环境。Castor JDO 封装了 Database、OQLQuery、QueryResults 3 种对象，这样业务层可以直接通过对象查询语言(Object Query Language)操作数据对象来实现数据库的操作。

(2)业务逻辑组件的实现

业务逻辑层是多层应用结构的核心，主要由一类后缀名为 Logic 的对象构成。

在这个类中通过 Castor JDO 操作对象来实现数据库的查询、新增、修改、删除等操作。而 Castor JDO 又是通过 OQL 查询语言来实现数据对象的操作。

OQL 查询通常用于在数据库中查找并创建一个 JDO 对象。OQL 查询语言类似于 SQL 查询，不过用对象名称而不是 SQL 名称并且不支持 join 子句。JDO 操作数据对象的实现方法如下：

1)创建 JDO 对象。

```
Database db = ( (JDO) GlobalVars.getValue(Constants.JDO_INSTANCE)).getDatabase();
```

2)对象操作。下列的代码片断使用 OQL 查询来将所有对象装载在一个给定的组中。注意，用户表是相关的对象，JDBC 查询包括了 join 操作：

```
UserTab user = null;
OQLQuery oql;
QueryResults results;
//建造一个新的查询并且设定其参数
db.begin();
oql = db.getOQLQuery(
"SELECT p FROM com.dc.sys.UserTab p WHERE p.userId = $1");
oql.bind(userId);
results = oql.execute(Database.ReadOnly);
//取回结果
while (results.hasMore()) {
user = (UserTab) results.next();}
```

查询执行 3 个步骤：1)使用 OQL 语句从数据库中创建查询 JDO 对象。2)如果有参数，那么该步骤包括设定这些参数。参数按相同的顺序出现在 OQL 语句中。3)执行查询和获得 org.exolab.castor.jdo.QueryResults 类型的结果集合。

查询能创建 1 次多次执行。每次执行完成，相关的参数自动消失，第 2 次查询必须再次提供。当查询第 2 次被执行时(有可能得到是不同的查询结果)，上次查询的结果(旧的)还能继续使用。

获得 User 对象，修改对象密码，并把修改的结果存储到数据库。新建一个 User 对象，并把对象存储到数据库。删除一个持久 User 对象(一旦提交数据库事务处理，该对象就会在数据库中被删除，同时在全部其它事务处理中消失)。

3.2 接口的实现

信贷管理系统当前主要从综合业务系统获取账务数据。贷款账务处理和结息由综合业务系统完成，账务数据以综合业务系统为准。信贷管理系统每天在工作日结束时通过数据接口批量获取综合业务系统的数据。

与综合业务系统的数据接口目前有 2 种接口方式：

(1)直接读取方式：直接读取综合业务系统数据库。这种方式的特点是需要与业务数据库连接，实现简单直接，数据采集效率高，但会影响业务数据库的效率。为了不影响综合业务系统的日常交易，并保证安全，数据读取安排在综合业务系统日终处理完成后批量进行。

(2)间接文本方式：信贷管理系统提供接口数据文本标准格式，综合业务系统在日处理过程中产生符合标准格式的文本数据，信贷管理系统以此文本数据作为数据源。这种方式的特点是信贷管理系统技术人员不需要了解综合业务系统数据库，信贷管理系统不与综合业务系统数据库直接相连，减少了对综合业务系统数据库的效率和安全性的影响，但对数据采集效率有一定影响。

为了提高处理效率和简化程序接口，本文采取第(1)种接口方式，但如果要强调安全性，可考虑第(2)种接口方式。

参考文献

1 Allen P, Bambara J. Sun Certified Enterprise Architect for J2EE Study Guide[M]. Osborne, McGraw-hill, 2003.

2 Ambler S W. Mapping Objects to Relational Databases, O/R Mapping in 33 Detail[EB/OL]. 2002-07. http://www.isys.uni-klu.ac.at/ISYS/Courses/03SS/eva/vounterlagen/5-XAmbler2000.

3 Allamaraju S. J2EE 编程指南(1.3 版)[M]. 北京: 电子工业出版社, 2002.

4 赵松涛. Oracle 9i 中文版数据库系统管理[M]. 北京: 人民邮电出版社, 2003.

5 梅登华, 闵华清. 基于 STRUTS 框架的 SRM 系统设计[J]. 计算机工程, 2006, 32(17): 262-265.

（上接第 255 页）

Express 提供了强大的后盾。基于 PCI Express 的主板已经上市，多家公司的多种基于 PCI Express 的显卡面世，Intel 公司正研制基于 PCI Express 的网卡和通用接口芯片，IDT 公司推出了 4 款由桥接和转换产品组成的 PRECISE 系列 PCI Express 产品，Xilinx 公司制造了支持 PCI Express 的集成 622Mbps～10.3125Gbps 串行收发器的 Virtex-4 FX60 90nm FPGA。可见在国外对于 PCI Express 的研发已经全面地展开，PCI Express 替代 PCI 势在必行。

6 结束语

众多产品的支持会令 PCI 总线的寿命延长，但 PCI Express 即将改变整个计算机系统结构、成为下一代总线标准的总线技术。在其应用方面，本文已经做了相关研究，下一步的工作是利用 FPGA 快速地实现 PCI Express 应用。

参考文献

1 Bhatt A V. Creating a Third Generation I/O Interconnect[EB/OL]. 2002. http://www.intel.com/technology/pciexpress/downloads/3rdGenWhitePaper.pdf.

2 Intel Corporation. PHY Interface for the PCI Express Architecture[EB/OL]. 2004. http://www.PSIG.com.

3 Intel Corporation. PCI Express Board Design Guidelines[EB/OL]. 2003. http://www.intel.com/technology/pciexpress/downloads/PCI_EI_PCB_Gui delines.pdf.