

用于数据访问的JavaBean自动生成器设计与实现

The Design and Realization of Auto JavaBean Generator for Data Access

摘要: 本文通过分析目前用于数据访问的JavaBean编写中所存在的问题,参考EJB的设计思想,提出了符合MVC设计模式的用于数据访问的JavaBean自动生成器,并详细地介绍了该自动生成器的设计思想与实现过程,最后演示了一个实例运行结果。

关键词: MVC模式 JavaBean 数据访问

李霖坤 (北京科技大学管理学院 100083)

彭岩峰 (北京中软金马计算机股份有限公司 100081)

1 引言

Java语言是当今最流行的一种跨平台的面对象的程序设计语言,尤其适合于基于Web的应用系统的开发,例如,电子商务、电子政务等。最近几年,在J2EE平台上开发的大型Web系统,大都采用模型-视图-控制器(Model-View-Control, MVC)的设计模式。但在很多中小型Web应用系统开发中,由于种种原因,Model数据层往往并不使用EJB组件来实现,这层通常使用软件开发人员自己编写的JavaBean来实现。事实上,大多数中小型项目的开发人员的主要精力就是编写这些JavaBeans。而这些JavaBeans的编写实际上是和数据库中的各个表相关联的,项目开发的主要工作就是拼写各种类型的SQL语句。如何才能将项目开发从这种重复性很强并且容易出错的工作中解脱出来呢?

针对这个问题,参考EJB的CMP实现思想,笔者设计了符合MVC设计模式,用于数据访问的可视化JavaBean自动生成器,使得以前需要花费大量时间和精力编写的JavaBeans,采用自动生成器,只需要几秒钟就可以轻松地自动生成。该文所介绍的JavaBean自动生成器的动态性、灵活性与易

用性不仅极大地减轻了软件开发人员的工作量,加快了项目开发进度,而且使程序易于维护。

2 JavaBean自动生成器的设计与实现

2.1 MVC模式

基于Web的应用系统的开发,由编程人员、数据库管理者、页面设计者、业务专家等各个方面的专业人员共同合作进行,此时如果各个组的工作之间依赖性比较高,工程项目的高效率进行就会变得比较困难。例如,Web页面的设计以及应用的逻辑设计如果密切相关的时候,无论开发作业还是测试,涉及变更都可能影响大范围的相关人员。所以人们想把整个工作分割为一个个低耦合的单元块,使得各个作业队伍能够分工协作。模型-视图-控制器(Model-View-Control, MVC)的设计模式就是在这种背景下提出的,并在大型Web应用系统开发中得到了广泛使用。

模型-视图-控制器是Xerox PARC在八十年代为编程语言Smalltalk-80发明的一种软件设计模式。最近几年被推荐为Sun公司J2EE

平台的设计模式,并且受到越来越多的Java开发者的欢迎。

MVC模式由三个部分组成。

(1) Model 数据层:应用系统的数据的存放。

(2) View 表示层:Model中存储数据的可视化表示。

(3) Control控制层:接受用户的输入,通知Model发生的事件。

在基于J2EE技术的Web应用系统开发中,JSP2的开发模式完全符合MVC的标准,JSP2模式中的三个组成部分(JavaBean/EJB, JSP, Servlet)分别对应了MVC的三个部分。

① Model 数据层:全部封装于Enterprise Java Bean控件中。

② View 表示层:由JSP负责处理页面的表示。如数据表示的格式,翻页,参照模式/编辑模式等等。

③ Control控制层:Servlet接受用户在页面的输入以及提交动作,并根据动作指示进行相应的业务处理(调用相应的EJB控件),然后根据处理结果交给相应的View表示层JSP程序,由它们负责表示。

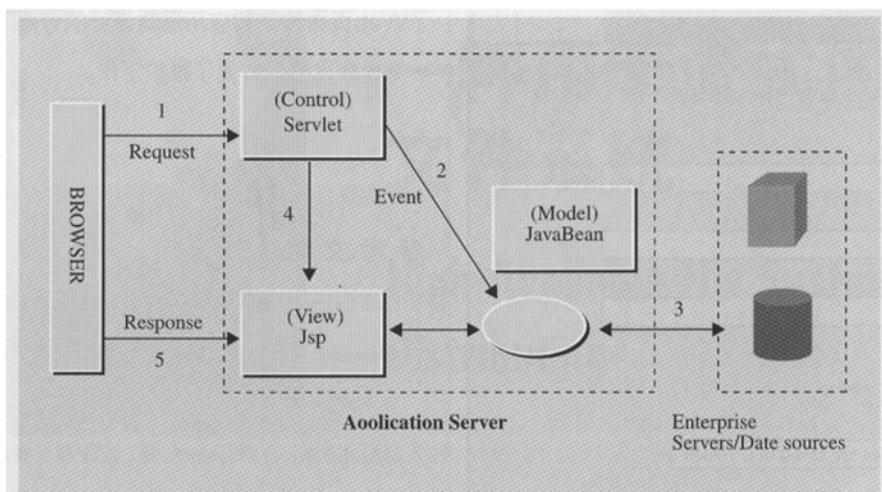


图1 MVC标准模式在JSP编程中的体现

图示说明:

1. Servlet接收HTML Form传送来的Http Request请求,并转换为Event事件。
2. Servlet根据相应的Event事件调用业务逻辑层的方法,进行业务逻辑处理。
3. Model层次的JavaBean/EJB通过JDBC/EJB进行数据库操作,并且将数据保存在JavaBean中。
4. Servlet取得对数据存放的JavaBean的参照以后,传递给JSP。
5. JSP根据数据,生成HTML页面,返回浏览器,并进行页面表示。

2.2 JavaBean 的介绍

JavaBean又简称为Bean,是Sun公司为了提高Java程序的可复用性而提出的基于Java的软件组件(Software Component)模型。所谓软件组件,是指提供程序员在开发程序时可直接应用的小程序。从功能上来说,可以利用Bean封装事务逻辑、进行数据库操作等,可以很好的实现业务逻辑和前台程序(如JSP文件)的分离。

从回调方法和Bean必须实现基本接口的观点看,Bean可以分为实体Bean和会话Bean。会话Bean可以看作是对服务器中客户端的一种扩展。实体Bean可以看作是数据库中数据的一种面向对象表示。本文所提到的JavaBean都只是处理与数据库操作相关的一些事务,如数据库表的插入、删除、修改等。

2.3 用于数据访问的JavaBean自动生成器的设计与实现

在很多中小型Web应用系统开发中,由于种种原因,Model数据层往往并不使用EJB组件来实现,这层通常使用软件开发人员自己编写的JavaBeans来实现。这种做法存在以下问题:

第一,项目开发人员的大部分时间和精力都花在编写相关的JavaBean上面,影响项目开发的质量和进度。

第二,如果业务需求变化,导致所访问的数据库表发生改变,项目开发人员需要检查并修改所有相关的方法,这个过程很容易出现错误或者遗漏。

第三,因为每个程序员的编程风格不一致,例如,有的JavaBean编写员在方法中获得数据库的连接,并将其作为成员变量使用,方法结束前自动释放连接资源;有的JavaBean编写员将数据库连接声明为全局变量,在对象使用完毕后,调用相关方法释放连接。这样导致JavaBean的最终调用者不知道是否需要手工释放数据库连接,给其带来许多麻烦,使程序难于维护。

为解决这些问题,笔者借鉴EJB的CMP设计理念,设计了用于数据访问的可视化JavaBean自动生成器,主要设计思想是针对所要访问的数据库表自动创建TableNameInfo类和TableNameMng类(在具体执行过程中,TableName将会被所要访问的数据库表的表名代

替)。TableNameInfo类实现了对所访问数据库表的所有字段的读方法和写方法。TableNameMng类封装了执行数据库表的CUIDS(创建、更新、插入、删除、查询)操作的SQL语句。在进行数据访问时,首先通过TableNameInfo类中的方法操作各字段的数值,然后将TableNameInfo类作为参数传递给TableNameMng类相应的方法,最终实现数据库表的CUIDS操作。这样免去了编程人员拼写SQL语句的工作,他们只需读取或写入各字段的数值,调用相应的方法就能实现数据库表的CUIDS操作。当数据库表发生变化时,只需运行JavaBean自动生产者,重新生成TableNameInfo类和TableNameMng类即可。

本文所介绍的JavaBean自动生成器通过RobotTableParse.java, RobotMng.java, RobotInfo.java, RobotGUI.java这四个类文件来实现,具体实现步骤是:首先运行类文件RobotGUI.java,用户选择要进行数据访问的数据库表文件,并利用类文件RobotTableParse.java对所选的数据库表文件自动解析,取出表名、主键及其数据类型、其余各列列名及其数据类型。然后根据所解析的表名、主键及其数据类型、其余各列列名及其数据类型,通过类文件RobotInfo.java, RobotMng.java自动生成该数据库表的TableNameInfo类和TableNameMng类。

2.4 实例演示

假设文件demo.sql的内容如下:

```

CREATE TABLE Departments_Table (
  Department_Code VARCHAR2(5)
  CONSTRAINT Dept_Dept_Code_PK PRIMARY KEY,
  Department_Name VARCHAR2(30),
  Department_Strength VARCHAR2(30));
  
```

运行上文所介绍的JavaBean自动生成器,运行结果如图2所示。该生成器将创建两个类文件:Departments_TableInfo.java和Departments_TableMng.java,默认情况下这两个类文件保存在桌面上。

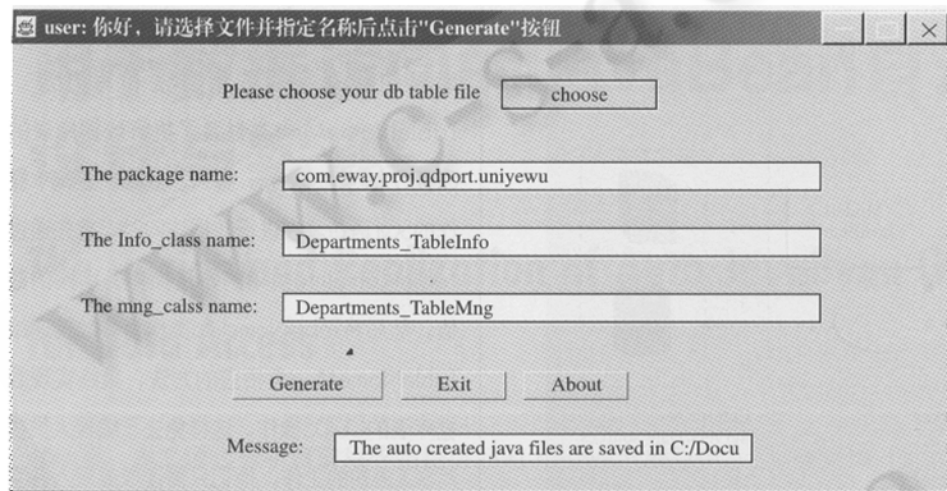


图2 JavaBean自动生产者运行界面

3 结论

通过在项目中的实际运用,证明了该文所介绍的JavaBean自动生成器不仅将项目开发人员从单调且容易出错的JavaBean编写中解脱出来,节省了项目开发所需的时间,而且有效地实现

了Web页面设计和应用逻辑设计的分离,统一了程序风格,方便了维护工作。

参考文献

- 1 Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 设计模式, 机械工业出版社, 2002年9月。
- 2 Subrahmanyam, Allamaraju 等, J2EE 服务器端高级编程, 机械工业出版社, 2001年9月。
Subrahmanyam, Allamaraju 等, J2EE 服务器端高级编程, 机械工业出版社, 2001年9月。
- 3 Bruce Eckel. Java 编程思想. 机械工业出版社, 2002年3月。