

J2EE 框架下 MIS 系统自动生成器的设计与实现

段其国, 郭剑毅, 时雷, 郭海, 杨彪, 江锦程

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650051)

摘要: J2EE 框架是构建现代企业级 MIS 系统的核心技术. 论文针对目前基于 J2EE 框架的 MIS 系统开发中自动化程度较低这一实际, 系统地进行了自动生成器的分析和设计, 并就系统实现的有关方法和技术进行了讨论. 其系统模型设计思想和实现技术已在实际的基于 J2EE 框架的 MIS 系统开发实践中得到应用, 取得了良好的效果.

关键词: 管理信息系统; J2EE; 自动生成

中图分类号: TP311.5 **文献标识码:** A **文章编号:** 1007-855X(2005)01-0037-04

Design and Realization of the Automatic Generator of MIS under J2EE Framework

DUAN Qi-guo, GUO Jian-yi, SHI Lei, GUO Hai, YANG Biao, JIANG Jin-cheng

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650051, China)

Abstract: J2EE Framework is the key technology to design MIS of modern enterprises. The analysis and design of the automatic generator are systematically carried on, and a discussion is made on the relevant methods and technology that realize the system aiming at the reality that the automatic degree is relatively low to develop MIS under J2EE Framework at present. And the model design and implementation technology have already applied in the development of MIS under J2EE Framework, which has achieved a good result.

Key words: MIS; J2EE; automatic generating

0 引言

管理信息系统是企业信息化建设的重要组成部分, 是提高企业效率和效益的关键环节. 随着 Internet 逐渐成为主要的信息处理平台和 J2EE 等新技术的出现, 构建在 J2EE 框架下的可跨平台、跨地域、具有灵活性、易维护性的基于 B/S 模式的管理信息系统(以下简称 J-Web-MIS)已成为 MIS 应用的趋势. 但目前对 J-Web-MIS 的开发仍没有一种高效的开发途径. 当前主要采用的方式是: 系统开发人员在充分掌握 J2EE 框架的基础上进行实体 EJB、会话 EJB, Servlet 和 Jsp 页面的手工编写. 这种开发方式其开发周期长、程序的质量得不到保证.

在这种情况下, 笔者提出利用 Java, Velocity 模板引擎和 XML 等技术, 通过分析和研究 J2EE 框架, 设计一个通用的 J-Web-MIS 生成器, 此生成器可以自动完成与数据库连接, 通过对数据库表信息的采集和分析自动生成实体 EJB, 而后生成会话 EJB, Servlet 和 Jsp 页面, 从而自动生成一个具有浏览、添加、更新和删除功能的 J-Web-MIS 系统. 这使得 J-Web-MIS 的开发效率得到很大的提高, 由于所有代码都是通过模板来产生的, 所以程序质量也得到保证.

1 J-Web-MIS 系统架构

J2EE 框架提供了一个基于组件的方法来设计、开发、装配以及部署企业应用程序, 组件主要有

收稿日期: 2004-04-16. 基金项目: 云南省自然科学基金资助项目(项目编号: 2002F0029M)和云南省信息技术项目(项目编号: 2002IT03).

第一作者简介: 段其国(1978-), 男, 在读硕士研究生, 主要研究方向: 模式识别与智能系统. E-mail: duangqg.cn@tom.com.net

Servlet, Jsp, EJB 等, 每个组件都在复杂的分布式应用系统中扮演一个特定的角色, 通过相互协调, 最终构成了基于 MVC 的具有高度灵活性和可重用性的企业应用体系. 图 1 即基于 J2EE 框架的 Web-MIS 系统的架构, J-Web-MIS 系统是在这个基本设计框架下搭建起来的.

Servlet 主要用于控制, 它接收来自客户端 Web 浏览器的 HTTP 请求, 处理输入参数, 再把结果以浏览器可以显示的 HTTP 方式进行回传, 对于用户操作数据库的请求, Servlet 通过调用 EJB 来完成. Jsp 用于显示 Servlet 的处理结果, 与使用者交互. EJB 是 J2EE 中间件中的分布式、可伸缩业务逻辑组件, 用于封装核心业务逻辑和数据模型元素^[1].

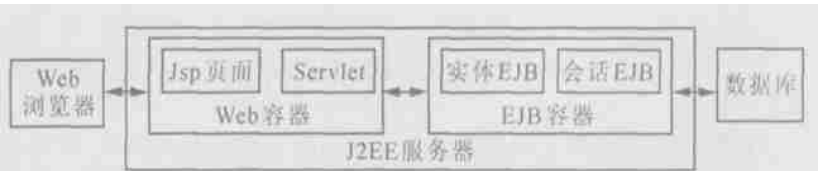


图 1 J-Web-MIS 系统架构
Fig.1 J-Web-MIS system structure

2 自动生成器的总体结构设计

为了保证 J-Web-MIS 自动生成器的质量, 自动生成器的结构采用分层式结构, 整体结构如图 2 所示. 自动生成器由“三层二接口”组成, 并提供对层和相应接口的管理.

1) “三层”. (1) 系统应用层: 包括界面视图管理和逻辑控制管理. 这一层主要实现了显示界面和用户交互, 接收用户录入的 J-Web-MIS 系统的描述信息, 包括: 单位名称, 开发者姓名, 版本号, 数据库类型等. (2) 自动生成层: 包括库表信息提取管理, 模板仓库管理, 实例生成管理和文件输出管理. 这一层实现了 J-Web-MIS 系统的自动生成, 是自动生成器的核心层. (3) 数据储存层: 用于存储数据库表信息.

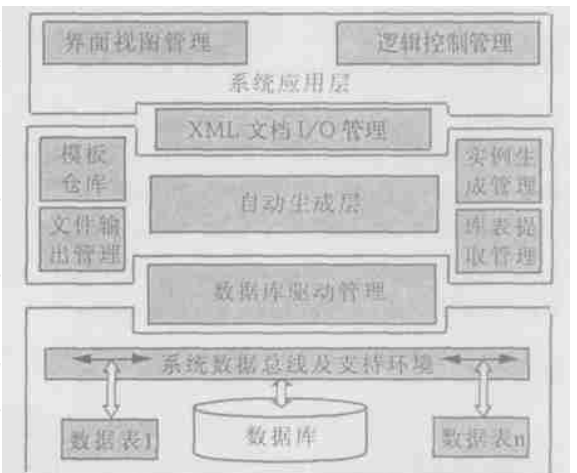


图 2 J-Web-MIS 系统自动生成器的总体结构图
Fig.2 Whole structure figure of J-Web-MIS automatic generator

2) “二接口”. XML 文档 I/O 管理接口: (1) 在系统应用层和自动生成层之间提供通信接口, 将用户在系统应用层录入的 Web-MIS 描述信息传递到自动生成层. (2) 数据库驱动管理接口: 在自动生成层和数据储存层之间提供通信接口.

3 自动生成器的实现

自动生成器的开发采用 Java 技术, 其原因有以下几点^[2]: (1) Java 语言是一种面向对象的编程语言, 便于实现代码的封装、继承和多态, 适合于自动生成器的构件化实现; (2) Java 语言本身的一些特性决定了用它开发出来的自动生成器具有良好的可扩展性和可移植性. (3) 模板引擎是自动生成器中的一个重要部分, Velocity 是一种功能强大的模板引擎技术, Java 语言可以方便的调用 Velocity.

自动生成器的功能主要由自动生成层来完成, 实现自动生成器的难点和重点在自动生成层的实现上, 下面对生成器的核心层-自动生成层作出论述. 自动生成层的工作流程示意图如图 3 所示.

3.1 模板仓库

模板仓库中存放着用于生成 J-Web-MIS 系统的模板. 对于 J-Web-MIS 系统中的实体 EJB 来说, 不同的 J-Web-MIS 系统其数据库表不同, 表中字段不同, 这使得各个 J-Web-MIS 系统中的实体 EJB 出现了差异. 但是, 实体 EJB 对数据库表的操作是相同的, 都是实现对记录的添加, 删除, 修改和更新, 这体现在代码上就是大量的代码相似.

例如, 一张部门表 DepartmentTable 具有两个字段: departmentid 和 departmentname, 与此表对映的实体 EJB 中创建一条记录的代码如下:

```

public Long.ejbCreate ( Long
departmentid,           String
departmentname) throws
CreateException {
    set Departmentid(depart-
mentid);
    set Departmentname(de-
partmentname);
    if( departmentid= null) {
        departmentid = U-
niqueIdGenerator. getInstance(). get
LongId(" DepartmentTable");
    }
    set Departmentid(depart-
mentid);
    return null;
}

```

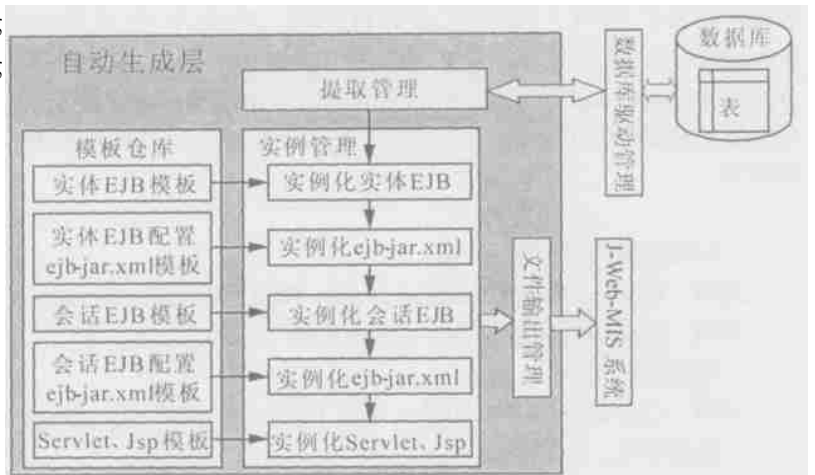


图 3 自动生成层的工作流程示意图
Fig.3 Workflow figure of automatic generating lay

斜体部分是根据不同的数据库表而变化的,可以看出这段代码中变化的只是一小部分.在会话 EJB, Servlet 和 Jsp 中也存在这种情况.究其原因是它们都要遵循 J2EE 规范,这就使得他们在基于不同应用时在结构和内容上都体现出共性.将上一段代码提取为模板,用 Velocity 模板引擎的语法表示,其片断如下:

```

public $ {entity. PrimaryKeyType}.ejbCreate($ {entity. RootPackage}. $ {entity. Name}Data value) throws CreateException {
    # foreach ($ field in $ entity. NonRelationFields)
        set$ {field. Name. Sentensized}(value. get$ {field. Name. Sentensized}());
    # end
    # if ($ entity. Relations. size() = 0)
        set$ {entity. Name}Data(value);
    # end
    # foreach ($ pkField in $ entity. PkFields)
    # if ($ pkField. HasAutoGenPrimaryKey)
    # if ($ pkField. Type. equals("Java. lang. Integer"))
        if (value. get$ {pkField. Name. Sentensized}() = = null) {
            .....
        }
    # end
    return null;
}

```

按照这种方法,开发出实体 EJB 模板 EntityBean. template、会话 EJB 模板 SessionBean. template、EJB 配置文件 ejb- jar. xml 模板 ejb- jar- xml. template, Servlet 模板 Servlet. template 和 Jsp 模板 Jsp. template. 这些模板文件都交由模板仓库进行管理和调用.

3.2 库表提取管理

库表提取管理通过数据库驱动接口与数据存储层进行通信.负责提取出数据库中每张表的信息,如表中的字段名称、类型、长度和是否为主键或外键等.库表提取管理、实例生成管理和文件输出管理等模块都

有相同的接口 JWM - Producer, 并通过 JWM - Producer 中的 initialization(JWM - ResourceManager, JWM - WrongManager) 完成各模块的初始化. JWM - ResourceManager 是资源句柄, 它配置各模块所需要的资源. JWM - WrongManager 是错误句柄, 它为系统应用层提供出错信息和出错处理. 这两个句柄和自动生成层的控制句柄 JWM - ConrolManager 都由自动生成层中的主控程序 Main 这个应用程序类来实例化. 代码片断如下.

```

.....
// 创建资源句柄、错误句柄
JWM - ResourceManager source = new JWM - ResourceManager();
JWM - WrongManager wrong = new JWM - WrongManager();
// 创建控制句柄
JWM - ConrolManager control = new JWM - ConrolManager(resource, wrong);
// 注册模板
control.setTemplate(" EntityBean. template"); // 注册实体 EJB 模板
.....
control.setTemplate(" Jsp. template"); // 注册 Jsp 模板
// 注册各模块
control.setProducer( TableAnalysis); // 注册库表提取管理模块
control.setProducer( FileOutput); // 注册文件输出管理模块
.....

```

代码中 control.setTemplate(" EntityBean. template") 是把实体 EJB 模板注册到控制句柄中, 以便实例生成管理模块调用. 同时, 控制句柄对自动生成层中的各个管理模块进行注册, 如 control.setProducer(TableAnalysis) 是把库表提取管理模块注册到控制句柄中, 类似地 control.setProducer(FileOutput) 是把文件输出管理模块进行注册. 其中, setProducer 方法实质上是把 JWM - ResourceManager 和 JWM - WrongManager 的实例赋给相应的管理模块, 从而完成了各管理模块的初始化. 注册后的各个管理模块可以通过控制句柄的 execute 方法来完成各自的功能.

3.3 实例生成管理

实例生成管理的初始化类似于库表提取管理. Velocity 模板引擎是实例生成管理的核心部分. Velocity 模板引擎首先通过库表提取管理生成的临时文件来对实体 EJB 模板进行实例化, 生成与数据表相映射的实体 EJB. 当实体 EJB 创建完成后, 其操作数据库表的方法名称也就确定下来了. 随之再调用会话 EJB 模板, 根据实体 EJB 中的方法名称来实例化一个会话 EJB, 这样会话 EJB 就可以通过调用实体 EJB 中的方法来操作数据库表. 在 EJB 生成后, 调用 ejb-jar.xml 模板, 根据 EJB 的信息来实例化 EJB 的配置文件 ejb-jar.xml 文件. 而后, 在根据会话 EJB 中的方法名称实例化 Servlet, 用于控制调用会话 EJB. 最后生成 Jsp 页面用于展示和用户交互.

3.4 文件输出管理

文件输入管理通过 JWM - Producer 接口程序来实例化, 使用 Java.io.* 包来将实例生成管理所生成的文件存放到用户指定的目录中. 生成的目录结构如图 4 所示.

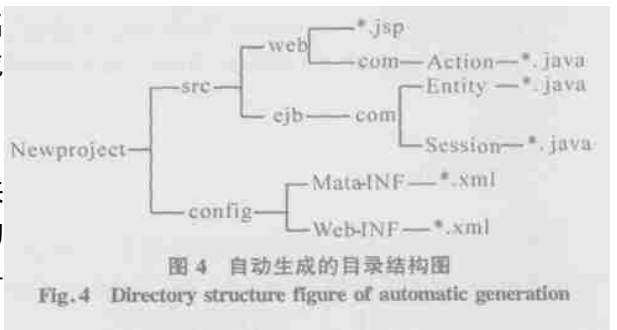


图 4 自动生成的目录结构图
Fig.4 Directory structure figure of automatic generation

(下转第 44 页)

$SNR = 0 \sim 1.2$, 一阶环路滤波器的阻带频率 $\omega_n = 100 \text{ Hz}$.

从图6和图7的对比可以看出环路的等效噪声带宽对VCO相位方差的影响很大,当二阶PLL等效噪声带宽为一阶PLL的0.2倍时,其VCO相位方差也为一阶PLL的0.2倍左右.一方面,环路带宽足够宽有利于VCO跟踪接收相位的任何时变;另一方面,宽带PLL允许更多的噪声进入环路,从而恶化相位估计.

5) 对平方环进行仿真,结果如图8所示.采样频率 $f_s = 1000$,载波频率 $f_c = 100 \text{ Hz}$, 噪信比 $1/SNR = 0 \sim 1.2$, 带通滤波器的通带频率为 $135 \sim 165 \text{ Hz}$, 环路滤波器的截止频率为 100 Hz .

6) 对科斯塔斯环进行仿真,结果如图9所示.采样频率 $f_s = 1000 \text{ Hz}$, 载波频率 $f_c = 100 \text{ Hz}$, 噪信比 $1/SNR = 0 \sim 1.2$, 两个环路滤波器的截止频率为 80.5 Hz , 低通滤波器的截止频率为 100 Hz .

由图8和图9对比可以看出,在噪声对VCO相位估计值的影响方面,科斯塔斯环和平方环的性能相似.

2 结论

从以上建模仿真结果的分析中我们可得出,在AWGN下噪声对平方环和科斯塔斯环相位估计的影响是相似的.事实上,如果科斯塔斯环中的环路滤波器与平方环中的环路滤波器相同,则二者是等效的.由于平方环的VCO输出信号的频率为输入信号频率的2倍,当输入载波频率较高时,VCO的倍频输出在工程上难以实现,所以此时一般采用科斯塔斯环;但平方环结构较科斯塔斯环简单,因此当条件受限且输入信号载波频率较低时可采用平方环.

参考文献:

- [1] [美] 普罗基斯. 数字通信(第四版)[M]. 张力军, 张宗橙, 郑宝玉译. 北京: 电子工业出版社, 2003. 247~ 260.
- [2] 张厥胜. 锁相技术[M]. 西安: 西安电子科技大学出版社, 2003. 62~ 66.
- [3] 谢嘉奎. 电子线路(非线性部分)[M]. 北京: 高等教育出版社, 1997. 378~ 388.
- [4] 李建新. 现代通信系统分析与仿真[M]. 西安: 西安电子科技大学出版社, 2001. 295~ 299.
- [5] 樊昌信. 通信原理(第四版)[M]. 北京: 国防工业出版社, 1995. 349~ 358.

(上接第40页)

4 结束语

论文提出了一种J-Web-MIS系统的自动生成器的解决方案,并着重描述了自动生成器的整体结构和一些关键部件的开发方法.J-Web-MIS自动生成器可以把编程人员从繁重的实体EJB,会话EJB,ejb-jar.xml,Jsp页面等实现上极大的解放出来.用户甚至可以不用学习J2EE规范,不懂Java语言,就能生成一个J-Web-MIS系统.当然,自动生成的J-Web-MIS系统功能是有限的,只限于浏览记录,添加记录,删除和更新记录等功能.但开发人员可在此基础上进行简单的二次开发就能实现一个功能完善的J-Web-MIS系统.因此J-Web-MIS自动生成器减轻了重复性工作,保证了程序的质量,极大的缩短了J-Web-MIS系统的开发周期.

参考文献:

- [1] SUN Microsystems. J2EE Overview[EB/OL]. <http://java.sun.com/j2ee/white.html>. 2001.
- [2] 董志斌, 吴启迪, 严隽薇. 一种基于Java技术的VHDL编译器的设计与开发[J]. 计算机工程与应用, 2000, 36(12): 75~ 76.
- [3] 李斌, 郭剑毅. 基于系统聚类的客户分析[J]. 昆明理工大学学报(理工版), 2004, 29(6): 66~ 69.