

劳动保障信息系统的设计与实现

贾艳宇

(珠海城市职业技术学院 工程与信息系, 广东 珠海 519090)

摘要: 随着互联网的快速发展,应用系统的架构从采用 C/S 结构逐渐转变到以 Web 应用为主。JAVA 和 XML 结合的方案应用在基于 Web 的信息系统开发,可以提高开发效率,改善软件质量,对基于 Web 的应用领域具有重要的实用价值。

关键词: XML; Java; 代码生成器; 信息系统

中图分类号: TP311.52

文献标识码: A

Design and implementation of the labor and social security IS

JIA Yan Yu

(Department of Engineering and Information, Zhuhai City Polytechnic College, Zhuhai 519090, China)

Abstract: With the developing of the Internet, the architecture of application system has been changed gradually from Client/Server to Web applications. The scheme of Java and XML techniques applied in Web-based information system development, can increase the efficiency of software development, improve software quality and may be utility value for Web application system development.

Key words: XML; Java; code generator; information system

随着劳动保障事业的快速发展,劳动保障业务信息的种类和信息量急剧膨胀,地区间信息交换与共享的需求日益迫切,信息安全管理需求与日俱增,现有的劳动保障信息系统已不能满足业务更替发展的需求。按照用户需求对原系统实现以下改造迁移:

(1)改造旧系统的管理架构,建立有效的业务和数据追踪机制;

(2)改造旧系统的信息存储机制,建立数据冗余和备份机制,系统中不再有数据物理删除;

(3)整合原菜单模块,建立新的功能管理机制,使各项业务适应业务变更、全市联网和分级管理的需要。

1 系统分析与设计

1.1 系统的技术方案设计

J2EE 平台提供了开发分布式、多层 Web 应用系统的标准 API,能够满足本系统的开发要求。本系统由 J2EE 技术实现,采用基于 XML 和 SOAP 的 Web Service 的 B/S 多层结构,选择具有多种成熟产品的关系型数据库 DB2。该技术方案的优点有:

(1)基于浏览器,客户端零安装、零维护,随处可用;

(2)J2EE 的多层企业级应用,为不同服务提供独立的层,易于扩展;

(3)分布式规模可变的的用户并发处理,利于信息交换和共享;

(4)DB2 高效数据层设计,支持海量的冗余备份数据存储。

1.2 系统硬件结构设计

系统在硬件上分为数据库服务器级、Web 服务器及应用服务器级、用户级 3 级。

数据库服务器级由数据库服务器和数据库灾备服务器组成。Web 服务器及应用服务器级由 3 台 Web 服务器及应用服务器机组并行运行。用户级的所有系统用户都通过所用 Internet、Intranet 访问服务器。

1.3 系统软件结构设计

系统的软件结构划分为表现层、业务层、数据层和存储层。为满足用户的需求,特别设计了 SHead、Rcdo、RC 类和 JobIntf、DConn 接口。

表现层用 JSP 实现,用于实现用户与系统之间的交互,并将用户的交互信息传递给业务层;业务层实现了

业务逻辑的封装,在这一层用户的交互信息均通过调用实现类 Rcd0 判断并调用相关业务实现类,各业务实现类统一实现了的业务实现接口 JobIntf 调用相应的私有业务实现方法,相应业务实现方法将用户请求转换为一系列具体的 SQL 语句,再通过 JobIntf 传递给数据层;数据层用数据接口类 Dconn 实现,接收来自业务层的 SQL 语句提交数据库执行并返回执行结果,反向传递回表现层。

系统保密类 SHead 存储用户关键信息,其对象在用户登录时生成并存储在用户会话中,将业务层传递给数据层,在 Dconn 中自动将其转换为数据库操作执行语句的一部分存入数据库。系统的层次结构如图 1 所示。

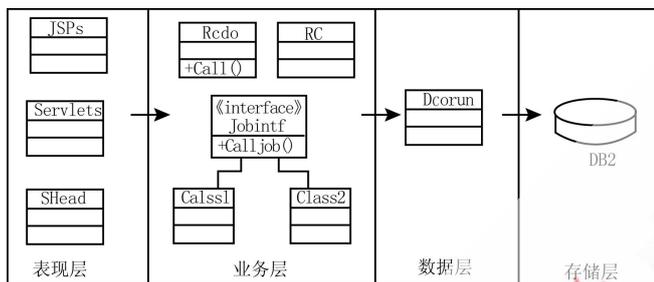


图 1 系统层次结构图

1.4 数据库设计

根据劳动保障业务的业务逻辑,本系统分为 11 个子系统,51 个模块。限于篇幅,在此仅给出系统管理子系统的数据库设计。该子系统必须满足下列需求:

- (1)维护用户的基本信息;
- (2)维护用户的权限申请、权限审批及分配信息;
- (3)维护密码及其修改的信息;
- (4)维护系统菜单及对应的程序源代码的操作信息;
- (5)维护数据库中表结构的信息;
- (6)维护数据库中执行 SQL 操作的信息;
- (7)维护系统参数及代码的信息;
- (8)维护系统权限设置的信息(系统权限的设置、删除、修改、查询及其分级、分组等)。

1.5 安全机制

本系统根据劳动保障业务的特殊需求,制定了严格的安全控制管理机制。

(1)业务全程追踪

从开发到实际运行阶段的维护工作,将所有人员的所有操作纳入系统的安全管理范畴,出现问题,可全程追踪事故原因。设置不同人员的接口,详细记录并严格控制各类人员的操作。

(2)权限管理机制

实行用户分类、用户权限分级、权限审批的管理机制,支持全市联网的数据共享机制。将用户权限分成超级用户、系统管理员、部门管理员、操作用户、访客用户;网上用户可查看系统权限,提出申请;新用户可用访

客身份在网上登录系统,申请用户名和权限;按照权限所属,系统提示部门管理员直接在系统上审批;系统管理员或部门管理员接到申请后,创建用户名及分配权限。根据业务管辖,使用权限控制区域数据的管理,同时将公众信息全面提供给访客。

(3)数据冗余备份机制

杜绝数据存储中的物理删除,只允许数据在业务逻辑上的屏蔽。将各阶段数据冗余备份在数据库中,真正做到故障的全面数据恢复。

1.6 代码生成器设计

在系统开发中,实现不同业务功能的代码有些是相似的,为了节约重复的代码工作,设计了一个代码生成器来自动生成 Java 程序源代码或代码的框架。代码生成过程如图 2 所示。



图 2 代码生成过程

2 系统的实现

仅以系统管理子系统机构部门管理模块为例,说明系统的实现。该模块用于超级用户和系统管理员进行系统机构的维护。

2.1 用户身份验证

用户的登录界面为 login.jsp,它调用 Loginvalidate.class 实现用户身份的验证:

```
<%@ page contentType="text/html; charset=GB2312" %>
<%@ page import="com.zh.lsspk.*"%>
<% session.invalidate();%>
<html><head><title>用户登录</title></head>
<body leftMargin = "0" scroll = 'no' topMargin = "0"
onload="win_onload()" background = "../img/back.gif">.....
<form id = "Form1" method = "post" action = "../servlet/
com.zh.lsspk.Loginvalidate">
.....
</form></body></html>
```

Loginvalidate.class 处理 HTTP 请求,用户的关键信息记录在 SHead 对象中:

```
public class Loginvalidate extends HttpServlet {
private static final String CONTENT_TYPE = "text/html;
charset=GB2312";
//处理 HTTP Get request
public void doGet(HttpServletRequest request, HttpServlet
Response response) throws ServletException, IOException {
doGet(request, response); }
//处理 HTTP Post request
public void doPost (HttpServletRequest request, HttpServlet
```

```

Response response) throws ServletException, IOException {
    response.setContentType(CONTENT_TYPE);
    PrintWriter out = response.getWriter();
    HttpSession session = request.getSession();
    .....
    try{ ... ..
//用户登录成功后保存敏感信息到 SHead 对象中
SH.setBmbm(UserDict.get("部门代码").toString());
SH.setBmmc(UserDict.get("部门名称").toString());
SH.setJgbm(UserDict.get("机构代码").toString());
SH.setJgmc(UserDict.get("机构名称").toString());
SH.setYhbm(UserDict.get("用户名").toString());
SH.setYhxm(UserDict.get("姓名").toString());
SH.setYhmm(UserDict.get("密码").toString());
...}

```

2.2 用户界面的实现

用户的界面用 JSP 实现。用户和系统的交互信息记录在 XML 文件中。如 FindconditionMenu.xml 存储了所有查找条件的信息,这些信息填充在用户界面的下拉列表中,以方便用户选择。

XMLTransform 类读取 XML 文档,并将其显示在下拉列表框中。下面显示了 XMLTransform.java 代码中相关的部分:

```

public class XMLTransform {
    private static XMLTransform transform = null;
    public XMLTransform() { }
    private Document ParseXMLFile(String xmlFile){
        DocumentBuilderFactory factory = null;
        DocumentBuilder builder = null;
        Document doc = null;
        try{
            factory = javax.xml.parsers.DocumentBuilderFactory.
newInstance();
            builder = factory.newDocumentBuilder();
            doc = builder.parse(xmlFile);
        }catch(Exception ex){
            System.out.println ("LoadXMLFile 错误消息:" + ex.
getMessage());
        }
        return doc; }
//读出 XML 文件并填充下拉框,并根据不同的值类型
//选用不同的限定条件
    public static String FillCondition (String xmlFile,String
FieldType,String valueChosed){
        StringBuffer OptionBuffer = null;
        NodeList optionNode = null; //作为下拉列表框中的节点
        try{

```

```

OptionBuffer = new java.lang.StringBuffer (""); //初始
化 StringBuffer
        transform = new XMLTransform();
        Document doc = transform.ParseXMLFile(xmlFile);
        Element root = doc.getDocumentElement(); //取得根节点
        //取得根节点属性并赋予数组
        String [] rootAttr = Util.spilt (root.getAttribute ("
AllConditionType"),",");
        //所有包含 Condition 的节点
        NodeList nodes = XPathAPI.selectNodeList (root,"//
Condition");
        for (int i = 0; i < nodes.getLength(); i++){
            Element eleNode = (Element)nodes.item(i);
            //获得 Condition 的 _type 属性并判断是否存在给定
            的类型
            for (int k = 0; k < rootAttr.length; k++){
                String K_RootAttr = rootAttr [k]; //第 K 个属性
                if (! FieldType.equals("")){
                    if (FieldType.equals(K_RootAttr)) {
                        optionNode = eleNode.getElementsByTagName
(K_RootAttr + "Option");
                    }
                }
                for (int j = 0; j < optionNode.getLength(); j++){
                    Element itemNode = (Element)optionNode.item(j);
                    String itemText = itemNode.getFirstChild().getNodeValue
();
                    String itemValue = itemNode.getAttribute ("
QuerySymbol");
                    String itemDefaultAttr = itemNode.getAttribute ("
IsDefault");
                    String optionSelected = "";
                    //判断选中的值是否为空,并设定选中项
                    if (! valueChosed.equals("")){
                        optionSelected = valueChosed.equals (itemValue)? "
selected" :"";
                    }else
                    { if (itemDefaultAttr.equals("Yes")){
                        optionSelected = "selected";
                    }
                }
            }
        }
    }
}

```

2.3 实现业务的类

实现机构部门管理业务的类为 Xt_Yhgl_jgbm.class,它实现了接口 JobIntf 的 CallJob()方法,通过该方法调用相应的私有业务实现方法,相应业务实现方法将用户请求转换为一系列具体的 SQL 语句,再通过 JobIntf 传递给数据层;数据层用数据接口类 Dconn 实现,接收来自业务层的 SQL 语句提交数据库执行并返回执行结果,反向



图3 机构部门管理页面之一

传递回表现层。该类实现对机构部门的增加、删除、修改、查看等业务操作。界面如图3所示。

2.4 数据模式的实现

该模块操作的表为“用户管理_机构信息”“用户管理_部门信息”。2个表的定义如表1和表2所示。实现代码略。

2.5 数据安全

Dconn 用于连接数据库和封装增、删、改等数据库操作。所有与数据库相关的操作都通过该接口来操纵数据

表1 用户管理_机构信息

主外键	字段	数据类型	说明	Null
PK	流水 ID	Bigint	流水 ID	
	部门 ID	Bigint	部门 ID	
	机构代码	Varchar(10)	机构代码	NULL
	部门代码	Varchar(10)	部门代码	NULL
	部门名称	Varchar(200)	部门名称	NULL
	部门地址	Varchar(200)	部门地址	NULL
	部门负责人	Varchar(80)	部门负责人	NULL
	总机电话	Varchar(40)	总机电话	NULL
	传真号码	Varchar(40)	传真号码	NULL
	电子邮件	Varchar(100)	电子邮件	NULL
	邮政编码	character(6)	邮政编码	NULL
	部门字	Varchar(10)	部门字	NULL
	操作机构	Varchar(10)	操作机构	NULL
	操作员	Varchar(80)	操作员	NULL
	操作日期	integer	操作日期	NULL
	操作时间	integer	操作时间	NULL
	记录状态	integer	记录状态	NULL
	备注	Varchar(1000)	备注	NULL

表2 用户管理_部门信息

主外键	字段	数据类型	说明	Null
PK	流水 ID	bigint	流水 ID	
	机构 ID	bigint	机构 ID	
	机构代码	Varchar(10)	机构代码	NULL
	机构名称	Varchar(200)	机构名称	NULL
	机构级别	Varchar(10)	机构级别	NULL
	机构地址	Varchar(200)	机构地址	NULL
	机构负责人	Varchar(80)	机构负责人	NULL
	总机电话	Varchar(40)	总机电话	NULL
	传真号码	Varchar(40)	传真号码	NULL
	电子邮件	Varchar(100)	电子邮件	NULL
	邮政编码	character(6)	邮政编码	NULL
	上级机构代码	Varchar(10)	上级机构代码	NULL
	机构字	Varchar(10)	机构字	NULL
	操作机构	Varchar(10)	操作机构	NULL
	操作员	Varchar(80)	操作员	NULL
	操作日期	Integer	操作日期	NULL
	操作时间	Integer	操作时间	NULL
	记录状态	Integer	记录状态	NULL
	备注	Varchar(1000)	备注	NULL

库,避免用户随意操纵数据。

2.6 代码生成器的实现

要实现代码生成器,首先是将某个具体业务模型转换为代码生成器可以读取的文件(.xml),然后代码生成器读取描述具体业务模型信息的 XML 文档,并依据定制的模板生成 Java 程序源代码(.java)。

(1)用 XML 文档记录和描述与技术无关的具体业务模型,XML 文档是代码生成器的输入文件。如果 XML 文档必须满足某种条件,可以在一个 XML 模式中编写数据约束条件。下面是一个简单的示范。

```
<?xml version="1.0"?>
<play name="Xt_Cdym_Ymgl_Ymmk">
  <Method name="Add" RC="rc" SHead="sh"/>
  <Method name="Edit" RC="rc" SHead="sh"/>
  <Method name="Del" RC="rc" SHead="sh"/>
  <Method name="Find" RC="rc" SHead="sh"/>
</play>
```

(2)XSLT 编辑器编辑包含业务模型的技术实现程序(XSLT 模板),通过 XSLT 处理器创建一个 XML 的语法树结构。图4是 XSLT 模板程序的相应的 XML 语法树结构。

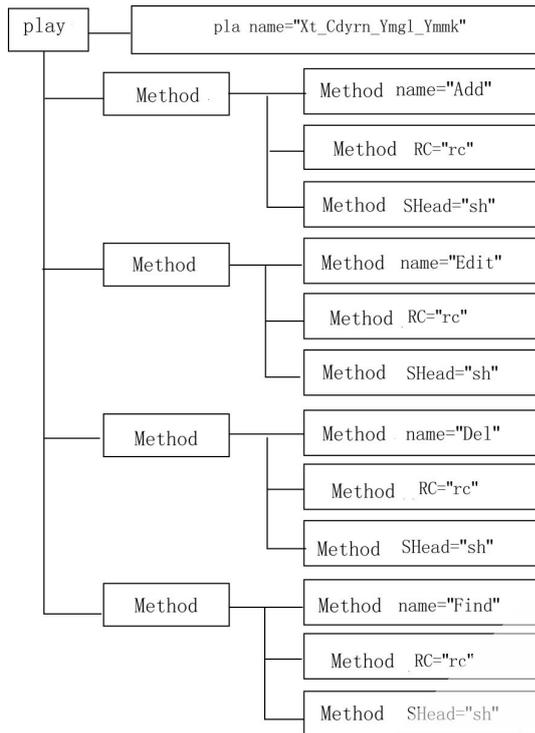


图4 example.xml 树结构

(3)XSLT 处理器获取 XML 文件的语法分析树,根据 XSLT 模板的脚本生成 Java 文件。这样,只要输入 XML 文件,XSLT 模板就可以结合输入的文件生成 Java 源代码。如有需求或技术更新,只需重新生成输入文件和模板文件,并用代码生成器再次生成源代码。

本系统采用 B/S 模式、XML 语言和典型的 N 层结构,以分离客户端、应用端和数据库服务器端,减少它们

相互间的耦合性。由于采用了 J2EE 框架结构,一旦业务量增长时,可以容易地重新设置部署,应付大量访问的需求。在需求变化时,也能够比较容易增加和修改组件,具有高度的可伸缩性和升级能力;同时由于 J2EE 框架结构和 Java 语言的优点,系统具有很好的安全性、稳定性和可移植性。

本系统所提出的代码生成的解决方案,可依据业务模型信息和各个业务处理逻辑的 XSLT 模板规则映射出 Java 源代码,该代码可直接编译运行。该方案依据用户的需求或技术的变化,快速生成有用的代码,提高软件开发的效率,对于企业和用户扩充和开发适合 Internet 环境下的应用有一定意义。

参考文献

- [1] 冯少荣.基于 XML 和 JAVA 构建程序生成器[J].计算机应用与软件,2005(1):57-59.
- [2] 杨绍方.深入掌握 J2EE 编程技术[M].北京:科学出版社,2002.
- [3] BAMBARA J J,ALLEN P R.J2EE 技术内幕[M].刘堃,译.北京:机械工业出版社,2002.
- [4] HERRINGTON J.Code-generation techniques for Java.http://www.onjava.com/pub/a/onjava/2003/09/03/generation.html.
- [5] 陈翔,王学斌,吴泉源.代码生成技术在 MDA 中的实现[J].计算机应用研究,2006(1):147-150.
- [6] 贾艳宇.XML 在信息系统开发中的应用[J].电脑学习,2009(4):30-31.

(收稿日期:2009-01-14)

作者简介:

贾艳宇,女,1966 年生,副教授,主要研究方向:数据库技术。

三剑侠笑傲江湖 酷睿 i5 独领风骚 ——新年装机大推荐

随着网络、高清电影、3D 游戏的不断发展,人们对于电脑性能的需求也在不断增加,性能强劲、性价比高,也是用户装机时永恒的追求。CPU 作为电脑中最为重要的部件,一直是大家装机时首要考虑的因素。2009 年,CPU 市场整体趋势是从双核往多核过渡,在双核已经逐渐普及后,多核 CPU 也已经走近用户身边。

进入 2010 年,眼下英特尔率先向市场推出超值新三剑侠系列处理器,包括智能钻石侠酷睿 i5-750 处理器,游戏铂金侠酷睿 2 四核 Q8300,网络白银侠奔腾双核 E6300。无论是双核还是四核,在游戏高清或者设计、影像制作等应用领域都有出色的发挥,不管是高端应用还是基本需求也都能有高性价比的表现。新年装机用英特尔新三剑侠,我看行!

(西岸奥美(北京)信息咨询服务有限公司供稿)