

基于 Struts 的民航地面运行监控系统的开发

The Development Of Ground Monitoring Airline System Based On Struts

李 飞 黄东军 (中南大学信息科学与工程学院 长沙 410083)

摘要:基于 Struts + JSP2.0 + EJB 的 Java Web 开发技术,是当今 Web 应用开发技术的主流,它不仅对 MVC 提供了很好的支持,而且也代表了最新的 B/S 信息系统开发模式。本文结合民航地面运行监控系统的实际开发过程,重点阐述了其核心技术及实际运用情况。

关键词:民航 JSP Struts EJB

1 引言

这套飞机地面运行监控系统是为民航地面保障系统各部门所设计的适用于组织、监控现场生产的信息系统。它通过航班动态信息、地面服务信息共享及加强生产进度信息的反馈,提高现场运行部门的监控生产进度的能力,保障航班正点,从而达到提高航班服务的质量。它可为民航系统各分子公司的飞机地面服务的各个环节提供支持,提高航班正常率,及公司航班服务质量。

好的 MVC 框架,很好地将应用逻辑、处理过程和显示逻辑分成不同的组件实现,对大型的复杂的 Web 系统而言,可大大方便代码开发和管理,提高开发效率。正是鉴于此特性,在设计民航地面监控系统时,我们采用了基于 Struts 框架的系统构建模式。

作为一个实现良好的 MVC 框架,Struts 对 Model、View 和 Controller 都提供了对应的实现组件,图 1 所示为 Struts 体系结构图。下面从视图、模型、控制来介绍 Struts 的体系结构。

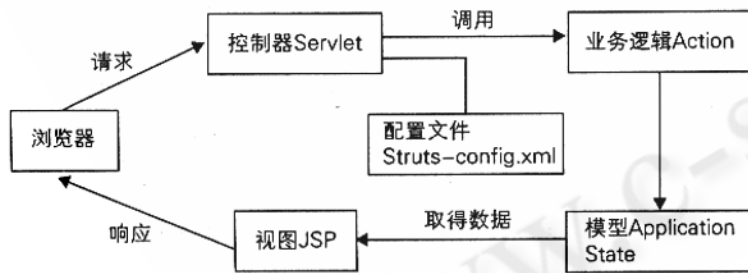


图 1 Struts 体系结构图

视图 (View): 主要由 JSP 生成页面完成视图, Struts 提供丰富的 JSP 标签库: Html, Bean, Logic, Template 等, 这有利于分开表现逻辑和程序逻辑。

模型 (Model) 表示应用程序的状态和业务逻辑, 通常由 JavaBean 或 EJB 组件来实现。

控制器 (Controller) 的作用是从客户端接受请求, 并且选择执行相应的业务逻辑, 然后把响应结果送回到客户端。在 Struts 中 Controller 由 ActionServlet 类和 Action 类来实现。ActionServlet 主要负责接受 HTTP 请求信息, Action 类负责调用模型的方法, 更新模型的状态, 并帮助控制应用程序的流程。

2 Struts 架构介绍

目前, 信息系统的开发方法很多, 但多数系统在开发中存在系统结构复杂、代码重用率低、网页设计和软件开发相互制约导致开发效率低下等不尽人意的地方。在系统设计过程中, 我们综合考虑系统结构、开发周期以及代码重用等方面, 决定采用先进的 J2EE 技术来开发这个应用系统, 并全面贯彻 MVC (Model - View - Controller) 的开发模式。而 Struts 作为一个实现良

3 系统设计与开发

3.1 开发环境

局域网环境, 数据源存于数据库服务器端, 服务器端的操作系统为 WINDOWS 2000 SERVER, 采用 ORACLE

数据库;应用程序位于 WEB 服务器上,WEB 服务器上安装的操作系统是 WINDOWS 2000 ADVANCE SERVER,应用服务器采用 BEA 公司的 WebLogic 7.0 作为 WEB SERVER。客户端操作系统为 WINDOWS 系列中文版,需安装 MICROSOFT 的 IE4.0 或更高版本。开发工具选用 JBuilder X。

3.2 功能模块

整个系统功能模块如图 2 所示。

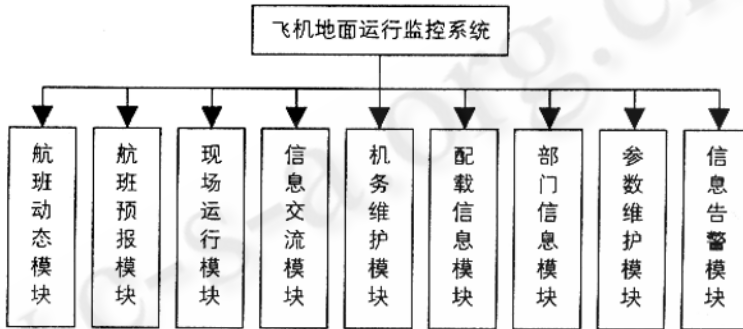


图 2 系统功能模块图

下面主要以的部门信息模块中的用户管理子模块为例,详细论述 MVC 各层次设计、开发过程及关键技术。

3.3 需求分析

部门信息模块的用户管理子模块主要是管理所有能登陆监控系统的用户的部门信息,记录其员工号、姓名、部门及基地信息,主要是针对数据库 eb_dept_admin_info(用户部门信息表)进行操作,其数据字典如表 1 所示。主要需求包括:新增用户部门信息,按姓名查找用户部门信息,修改用户部门信息,删除用户部门信息,是一个典型的信息系统的基本需求。

表 1 eb_dept_admin_info 用户部门信息表

字段名	类型	约束	说明
staff_num	varchar(12)	Not NULL	员工号
chn_name	varchar(20)	Not NULL	员工姓名
dept_id	varchar(20)	Not NULL	部门名称
base	varchar(3)	Not NULL	基地代码

3.4 流程设计

部门信息模块中的用户管理子模块的流程设计如

图 3 所示,该流程设计主要依据 Struts 的 MVC 模型建立。图中 JSP 文件及对应的 ActionForm 为视图文件(View)、Action 为控制器文件(Controller)。模型文件(Model) Dao 因为版面的原因没有画出,他们都是在 Action 被调用,用来访问数据库,进行持久化层操作。

3.5 视图开发(View)

视图开发主要是 JSP 页面开发及对应的 ActionForm 表单开发。用户管理子模块中开发的 JSP 页面

有: deptInfoAdminAdd.jsp, 新增用户部门信息界面; deptInfoAdminUpd.jsp, 修改用户部门信息界面; deptInfoAdminList.jsp, 用户部门信息显示界面(包括查询); deptInfoAdminResult.jsp, 用户操作结果提示界面。

ActionForm 是一种用于在视图组件和控制器组件之间传递 HTML 表单数据的特殊的 JavaBean。本模块所需使用的 ActionForm 只有一个,就是 deptInfoAdminForm,用来存储员工号、姓名、部门、基地等信息。它包括以下几个字段: staffNum, chnName, deptId, base, oldStaffNum(主键可能更改,用来标志原主键值), queryChnName(用来查询数据库所用的员工姓名值)及与它们对应的 set, get 方法。

另外, ActionForm 的 validate() 方法可用于实现对用户输入数据进行格式和语法检查,但是实际开发中这种方法并不有效;通常我们还是在 JSP 文件中嵌入 JavaScript 来对输入数据进行合法性检查。关于 JSP 页面、ActionForm 与 Action 的交互过程在这里不便赘述,在这里只是想强调的是:系统在处理用户的 JSP 页面(包含有 <html:form>)请求的时候,会先根据配置文件的信息查找所需的 ActionForm 是否有实例。如果没有,则会创建新的实例;如果有,则其不为 null 的属性

值会返回到用户所申请的页面,与 JSP 页面中相应的 Struts 标签的值相对应(比如 <html:text>, <html:select>等)。这一点在处理修改页面的时候非常有用(在后面会进行详细介绍)。

视图开发中最主要的是信息显示界面的开发,即: deptInfoAdminList.jsp。该页面是 deptInfoAdminAction.do? method = selRecord (Action) 的返回页面,包含着查询数据库结果集列表 (deptInfoAdminVo 组成的列

使应用更加便于维护,我们扩展 DispatchAction 类来完成一组相关的业务操作。

与扩展普通的 Action 类不同,创建一个扩展 DispatchAction 类的子类,不必覆盖 execute() 方法,而是创建一些实际操作的方法,这些业务方法都应该和 execute() 方法具有同样的方法签名,即他们的参数和返回的类型都应该相同,此外也应该抛出 Exception。实际开发的 DispatchAction deptInfoAdminAction.java 的

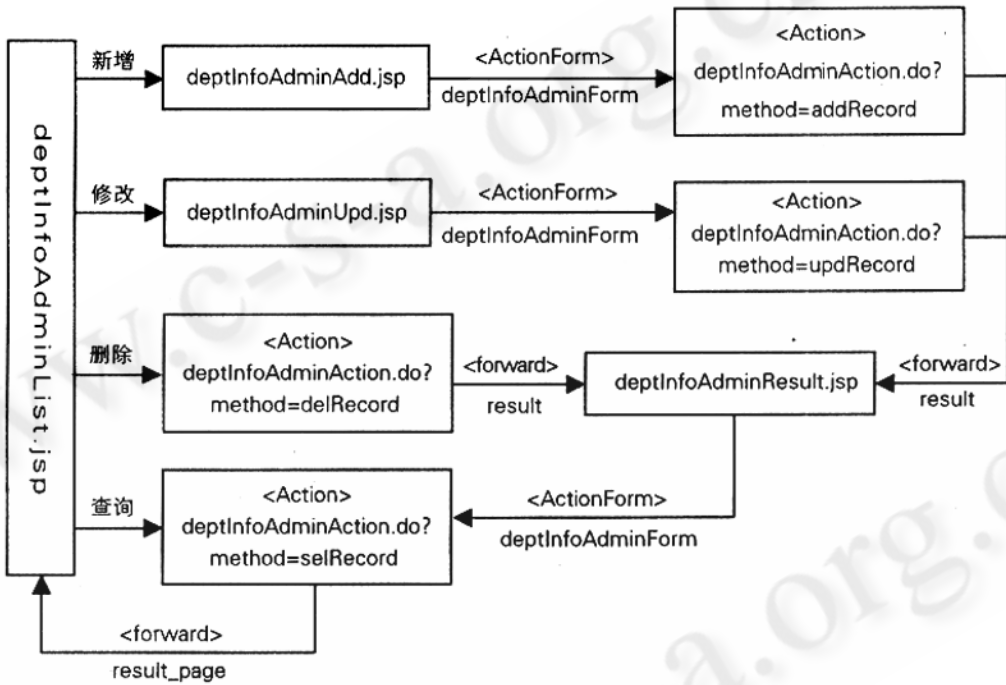


图 3 用户管理子模块的流程设计图

表)。在页面中,一般只需要遍历 deptInfoAdminVo 列表,就可获得需要的结果列表。实际开发中,考虑美工及分页的因素,我们采用一个成熟的标签控件 DisplayTag。DisplayTag 是一个非常好用的表格显示标签,适合 MVC 模式(DisplayTag 的使用说明在这里不加以赘述)。

3.6 控制器开发(Controller)

控制器组件包括 ActionServlet 类和 Action 类。ActionServlet 类是 Struts 框架自带的,它是整个 Struts 框架的控制枢纽,通常不需要扩展。Struts 框架提供了可供扩展的 Action 类,它用来处理特定的 HTTP 请求。

通常,在一个 Action 类中只能完成一种业务操作,如果我们需要完成新增、修改、删除、查找信息等各项操作,我们需要分别编写相应的各种 Action 来完成对应的业务需求。实际开发中,为了减少重复编程,同时

部分代码如例程 1 所示。

例程 1 deptInfoAdminAction.java

```

public class deptInfoAdminAction extends BaseDispatchAction {
    public deptInfoAdminAction() {
    }

    public ActionForward addRecord ( ActionMapping mapping, ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws
        Exception {
        deptInfoAdminForm deptInfoAdminForm =
        (deptInfoAdminForm) form;
        String title = null;
    }
}
    
```

```

String identify = null;
try {
    deptInfoAdminDao dao = new deptInfoAdminD-
ao();
    boolean result = dao.addDeptAdmin ( deptin-
foAdminForm );
    if ( result ) {
        title = " 增加成功,请确认。";
        identify = " success_add";
        request.setAttribute ( " title", title );
        request.setAttribute ( " identify", identify );
    }
    else {
        title = " 增加失败,请检查输入数据。";
        identify = " fail_add";
        request.setAttribute ( " title", title );
        request.setAttribute ( " identify", identify );
    }
}
catch ( Exception e ) {
    e.printStackTrace ();
}
return mapping.findForward ( " result" );
}
...
}

```

以上方法包括了业务需求中的新增、查询、修改及删除全部操作。新增用户部门信息的方法 (addRecord 方法) 的全部代码已经给出, 代码中的 deptInfoAdminForm 就是 Action 所对应的 ActionForm, 用来传入用户界面上输入的参数; deptInfoAdminDao 是模型组件 (在后面将做介绍), 用来访问系统 Oracle 数据库, 在相应表中增加记录。无论操作成功与否, 都将对用户返回 deptInfoAdminResult.jsp 页面 (Struts 配置文件中 < forward > 属性的配置值), 同时显示反馈信息。调用该方法, 只需要将相应的 JSP 文件的 < html:form > 的 action 属性值设为: deptInfoAdminAction.do? method = addRecord, 同时在 Struts 配置文件中将 < action > 配置增加 parameter = "method" 的属性即可。

3.7 模型开发 (Model)

模型组件代表系统应用的业务逻辑, 它包含了业务实体和业务规则, 负责访问和更新持久化数据。持久化意味着通过手工或其他方式输入到应用中的数据, 能够在应用结束运行后依然存在; 实际开发中所使用到的持久化数据一般是指的关系型数据库。

软件开发中, 面向对象的开发方法是当今的主流, 同时又需要大量使用关系型数据库, 因此在实际开发中, 对象 - 关系的映射 (Object - Relation Mapping, 即 ORM) 是一种棘手的操作。围绕对象 - 关系的映射和持久化数据的访问, 我们使用了一种软件领域发展起来的数据访问对象 (Data Access Object, 即 DAO) 设计模式。这种模式提供了访问关系型数据库系统所需的所有操作的接口, 其中包括创建数据库、定义表、字段及索引, 建立表间的关系, 查询、更新数据库等。

在这里我们主要是对 Oracle 数据库的用户部门信息表 (eb_dept_admin_info 表) 进行查询、插入、修改及删除操作, 所创建的模型组件 deptInfoAdminDao.java 的部分代码如例程 2 所示。

例程 2 deptInfoAdminDao.java

```

public class deptInfoAdminDao {
    public deptInfoAdminDao () {
    }
    public boolean addDeptAdmin ( deptInfoAdmin-
Form form ) throws DAOException {
        boolean success = false;
        Connection conn = null;
        PreparedStatement stmt = null;
        try {
            String addSql = " insert into eb_dept_ad-
min_info values ( ? , ? , ? , ? ) ";
            conn = DbConnectionProvider.getEBCon-
nection ();
            stmt = conn.prepareStatement ( addSql );
            stmt.setString ( 1, form.getStaffNum ( ) );
            stmt.setString ( 2, StringUtil.encodeStr ( form.getChn-
Name ( ), " GBK", " ISO - 8859 - 1" ) );
            stmt.setString ( 3, form.getDeptId ( ) );
            stmt.setString ( 4, form.getBase ( ) );
            stmt.executeUpdate ();
            success = true;

```

```

    }
    catch ( SQLException ex ) {
        throw new SQLException( ex );
    }
    catch ( Exception ex2 ) {
        throw new DAOException( ex2 );
    }
    finally {
        DbUtil. closeStmt( stmt );
        DbUtil. closeConn( conn );
    }
    return success;
}
...
}

```

以上方法包括了对数据库表的插入、查找、修改、删除的所有操作,其中插入数据库表的 addDeptAdmin 方法的代码已经全部给出。该方法在例程 1 中被 Action 的 addRecord 方法调用,用来对 eb_dept_info_admin 表进行插入操作,实现业务逻辑上的新增用户信息需求。方法中调用了创建数据库连接的 JavaBean (DbConnectionProvider. java) 的 getEBConnection () 方法创建数据库连接,使用相应的 SQL 语句进行数据库操作,最后返回插入操作是否成功的 boolean 值。逻辑非常清楚,给今后维护人员的代码阅读及修改带来了极大的方便。

3.8 EJB 组件开发

实际开发中,系统所需要调用的一部分数据来自民航系统的内部数据库(例如:机组成员信息表、航班运行时刻表、配载舱单信息表等),出于安全性的考虑,这些数据不能发布到外部的数据库服务器,也不能随便被修改、删除等以免造成生产中的巨大损失。

同时,系统还需要调用一些民航系统内部的业务逻辑(例如:飞行小时的计算,飞行员的排班等),这些业务逻辑是极为重要的公司财产,具有很高的安全性要求,也必须与外部的 Internet 隔离开。

对于以上需求,我们需要开发相应的 EJB 组件,部署在 Bea Weblogic 8.1 平台中。出于篇幅的考虑,在这里就第一种情况进行简单介绍。

为了访问内部的数据资源,我们编写了名为 ejbd-

bAccess 的 EJB 组件,是一种无状态的会话 Bean。其 Remote 接口文件为 EJBDBAccess. java, Home 接口文件为 EJBDBAccessHome. java。其 Enterprise Java Bean 类文件为 EJBDBAccessBean. java,为 EJB 核心文件,代码略。

3.9 Struts 框架配置

正如前面提及的,Struts 框架允许把应用划分成多个组件,提高开发速度。而 Struts 框架的配置文件 struts - config. xml 的作用就是把这些组件组装起来,决定如何使用它们。

4 总结

Struts 是一个实现良好的 MVC 框架,在现代软件开发中,越来越多的应用于大型 Web 应用系统的开发。基于 Struts 框架的民航地面运行监控系统的顺利开发,也证实了 Struts 框架是一种非常优秀的 J2EE/ MVC 实现方式。使用 Struts,开发人员不用再花大量时间和精力用来设计、开发自己的框架,可以集中时间精力在如何解决实际业务问题上。

Struts 框架是一群经验丰富的 Web 开发专家的集体智慧结晶,在全世界范围内得到了广泛应用及一致认可。使用基于 MVC 开发模式的 Struts 架构,可以使软件在健壮性、代码重用和结构方面上一个新的台阶;同时也为后期的系统维护、升级提供了极大的便利。以上这些优点,我们在民航地面运行监控系统的开发过程中深有体会。

基于 Struts 的民航地面运行监控系统的顺利开发,提高了现场运行部门的监控生产进度的能力,保障了航班正点,提高了航班服务的质量。它为民航系统各分子公司的飞机地面服务的各个环节提供了极大的支持,提高了航班正常率及公司航班服务质量,对于中国民航事业的发展具有重要意义。

参考文献

- 1 孙卫琴,精通 Struts:基于 MVC 的 Java Web 设计与开发,电子工业出版社。
- 2 Chuck Cavaness, JaKarta Struts 编程,清华大学出版社。
- 3 飞思科技产品研发中心,JSP 应用开发详解,电子工业出版社。
- 4 Hans Bergsten. JSP 设计. 中国出版社。