

基于组件的电厂设备检修系统分析与设计

李珊, 黄东

(东南大学自动控制系, 江苏南京 210096)

摘要:组件和基于组件的软件开发方法是管理信息系统革命的驱动力。文中简略讨论了组件、基于组件的软件开发方法以及 .Net 这种组件开发平台, 并介绍了组件技术在电厂设备检修系统分析与设计中的应用。基于组件的软件开发方法实现了软件的复用, 提高了软件开发的效率, 节省了软件开发费用并且易于扩展。应用这种方法开发出的电厂设备检修管理系统已经安全可靠地应用于某电厂生产中, 大大提高了电厂设备检修的管理水平。

关键词:组件; 基于组件的软件开发方法; 电厂设备检修管理系统

中图分类号: TP311.52

文献标识码: A

文章编号: 1005-3751(2006)02-0059-03

Analysis and Design of Power Plant Equipment - Repairing System Based on Component

LI Shan, HUANG Dong

(Automation Control Department, Southeast University, Nanjing 210096, China)

Abstract: Component and component-based development(CBD) bring a revolution to manager information system(MIS). The article discussed component, CBD and .Net, a developing platform of component, and introduced the using of CBD in power plant equipment-repairing system. CBD improves the reuse of software, and makes the development of software more efficient, more economical and can be easily expanded. The power plant equipment-repairing management system which developed in CBD has been used safely in the work of a power plant, and improves the management of equipment-repairing markedly.

Key words: component; CBD; power plant equipment-repairing management system

0 引言

电厂的管理方法是不断发展和变化的。电厂的设备检修管理系统必须达到以下目标:控制复杂性、短发布时间、易于重构、提高一致性、提高生产率、支持分布式开发、减少维护费用。要达到以上目标,传统的结构化方法和成熟的面向对象方法是不能胜任的,组件和基于组件的软件开发方法(Component-Based Development, 简称 CBD)提供了最有希望的尝试。文中简要讨论了基于组件技术的软件开发方法,介绍了组件技术在华能南通电厂设备检修管理系统开发中的应用。

1 CBD 及其相关概念

1.1 组件

一个软件的组件是可执行软件的一个可分离的部分^[1],是一个标准的可以互换、装配完成的软件模块的封装部分。组件的重要特征是接口和实现分离。组件是一个可交付的软件单元,它提供一些有用的功能和服务,满

足用户的需要。组件对外界提供统一的接口,外界无法访问组件的内部结构。组件通过接口提供服务,客户无法看到组件的内部实现。可以认为组件是一种封装对象的简单方法,被封装的对象在组件标准下实现功能,它们通过组件接口对外提供服务^[1]。然而组件是有别于对象的,组件的要素主要有:规格说明,组件标准,包装与部署,存储在组件库中的组件,基于接口的系统设计,基于组件框架的应用系统组装。

1.2 基于组件的软件开发方法(CBD)

CBD 是一种利用可重用的软件组件构建应用程序的技术,简单地说就是通过一些预先构建的软件组件来组装成新的软件,而不是从头开发^[2],这种组装是基于那些以独立服务形式开发出来的组件进行的^[3]。CBD 所从事的首要活动就是收集、创建和组装组件,而随之也产生了一些其他的主要活动,包括改善组件质量,使组件之间能够相互适应,以及更新组件。CBD 的基本目的是向以组装方式进行应用程序开发的方向发展。这种组装是基于那些以独立的服务开发出来的组件进行的,这些组件的服务是通过在某种通用的基础设施上调用某种服务来进行信息的交互的。由此可见基于组件的软件开发所必须的 4 种关键要素:由组件组装的应用程序,独立服务提供,公共组件基础设施,标准服务的使用^[4]。

收稿日期:2005-07-19

作者简介:李珊(1977—),男,陕西户县人,硕士研究生,研究方向为计算机管理与控制综合自动化;黄东,副教授,主要从事控制理论与控制工程教学与研究。

1.3 .NET 平台

前面讨论了组件及其基于组件的软件开发方法与技术,但是要在软件项目中实施组件技术,还必须采用一种具体的组件技术。当前可以使用的几种颇受欢迎的、并具有一定竞争力的企业应用程序开发平台包括 CORBA、.NET 和 Java2 Enterprise(J2EE)。

.NET 是一种面向网络、支持各种用户终端的开发平台环境^[5]。.NET 框架具有两个主要组件:公共语言运行库和 .NET 框架类库。公共语言运行库是 .NET 框架的基础,作为一个在执行时管理代码的代理,它提供核心服务(如内存管理、线程管理和远程处理),而且还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准确性。NET 框架类库是一个综合性的面向对象的可重用类型集合,可以用于开发包含从传统的命令行或图形用户界面(GUI)应用程序到基于 ASP.NET 所提供的最新创新的应用程序(如 Web 窗体和 XML Web services)在内的应用程序。

.NET 提供了一个标准的组件交互接口 IComponent,实现 IComponent 接口的类就形成了 .NET 组件。这样的接口标准允许开发人员将组件快速插接在一起以创建较大的程序。NET 框架提供了 IComponent 接口和 Component 基类,使创作适合设计环境的组件更为容易;还提供了 UserControl 和 Control 类,使创作可视组件也更加容易。Component 类是 IComponent 的默认实现,是公共语言运行库中按引用封送的所有组件的基类。

2 华能南通电厂设备检修系统分析与设计

华能南通电厂设备检修系统是针对电厂设备检修管理这一领域,经过详细的需求分析,设计开发出设备检修领域的组件系统,并利用该组件系统组装成为一套设备检修系统。

2.1 系统分析

机组检修管理的总体功能视图如图 1 所示。

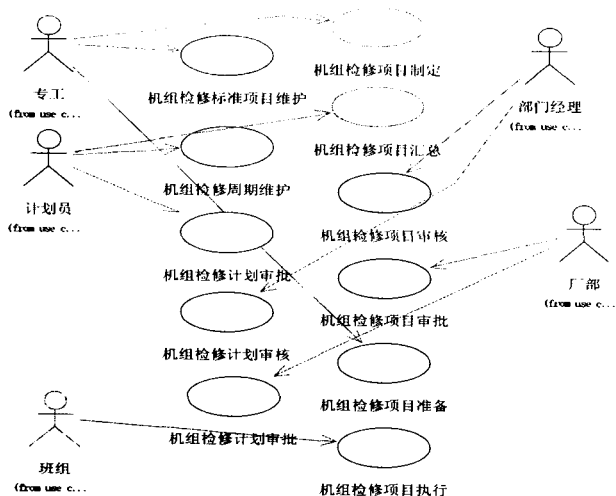


图 1 机组检修管理总体功能视图

设备检修系统分为设备维护管理、公用系统及辅机检

修管理、机组检修管理等子系统。文中只介绍机组检修管理子系统。机组检修周期管理包含要检修的机组、检修等级、检修周期以及上次检修日期。在机组检修标准库包含机组检修的标准项目,每个标准项目包含有子项目、设备列表、工艺卡、记录表、备品材料以及工具等信息。由检修周期和上次检修日期得到机组检修计划。由机组检修计划以及机组检修标准库生成机组检修项目。通过对机组检修项目的制定、准备以及执行 3 个流程,完成机组检修计划。

2.2 组件系统设计

组件系统采用层的体系结构模式,将组件系统的设计分为数据访问层组件设计、业务逻辑层组件设计和表示层组件设计。数据访问层组件由 Core 组件和 DbConfig 组件实现,其中 DbConfig 是数据访问的一些动态参数的 XML 配置。Core 完成对数据库的直接访问,其它业务逻辑组件均通过 Core 间接访问数据库,这样使得业务逻辑组件不会依赖某个特定的数据库。不同的数据库只需要提供不同的 Core 就可以。业务逻辑层才是组件系统真正的业务中心,包含有 SunitRepairPlan(机组检修标准库组件),PUnitRepairPlan(机组检修计划组件),Common(公共信息组件)。在表示层,设计了 MyBasePage(公共页面基类组件)。各个组件的功能以及相互依赖的关系见表 1。

表 1 组件功能及依赖关系表

| 编号 | 组件名称 | 作用 | 需引用组件 | 被引用组件 |
|----|-----------------|--|-------|-------|
| 1 | DbConfig | 业务逻辑组件的数据库连接配置操作 | None | 2 |
| 2 | Core | 数据访问公共基础类库,提供数据库访问的高级方法 | 1 | 4,5,6 |
| 3 | Common | 公共信息服务组件,可以得到专业、班组信息、用户名的信息等 | 其他 | none |
| 4 | MyBasePage | 公共页面基类。封装页面的出错处理以及各个页面公用的一些页面逻辑 | 2 | none |
| 5 | SunitRepairPlan | 机组检修标准库组件,包括机组检修周期的维护,机组检修标准项目维护以及标准项目对应的文件包维护 | 2 | 6 |
| 6 | PUnitRepairPlan | 机组检修计划组件,包括年度计划制定,机组检修计划制定,机组检修计划执行 | 1,2 | none |

(注:其他是指其他子系统或者模块的业务逻辑组件引用)

组件的详细设计主要从组件的静态结构类图和它所提供的接口来说明。详细介绍 SunitRepairPlan 组件和 PUnitRepairPlan 组件,其他从略。

数据库配置组件(DbConfig),主要就是配置数据库连接字符串 connectionString 和机组计划制订中用到的相对于当前年度的年数 numberOfYear。ModuleSettings 类封装了配置参数,MDbConfig 主要是两个静态函数 GetSettings()得到配置参数,SaveSettings()保存设置。数据访问的公共组件(Core),包含数据库访问类(DbObject),所有访问数据库的类都从 DbObject 类继承下去,这样数据库访问的所有操作都使用 DbObject 的函数,程序中全部

使用 sql 语句。公共信息服务组件(Common)主要是对其他组件提供公共信息服务,主要是得到用户信息、专业信息、班组信息,根据项目实际的需要,采用的适配器设计模式。适配器设计模式将一个类的接口转换成客户希望的另外一个接口,使得原本不兼容而不能一起工作的那些类可以一起工作。

机组检修标准库组件(SunitRepairPlan),主要是对机组检修周期和机组检修标准项目及其项目相关的一些东西进行维护和封装。这里采用了会话外观的设计模式。结构图如图2所示。由图可见机组检修标准主要通过 SunitRepairStandard 和 SunitCycle 对外提供接口。SunitCycle 对应机组检修周期; SunitRepairStandard 对应机组检修标准项目, SunitRepairStandard 提供了操作检修项目本身和与标准项目相关的一些信息(工艺卡、备品材料、工器具等)的接口,这是会话外观设计模式的运用。

机组检修计划组件(PUnitRepairPlan)类结构如图3所示。

组件的设计思想与机组检修标准库相同,采用会话外观的设计模式。提供接口的类是机组检修计划操作 PUnitYearPlan 和机组检修项目操作 PUnitRepairItem。同时 PUnitRepairItem 作为外观封装了对项目本身的操作、子项目的操作、设备列表的操作、所需工器具和备品材料等操作。

2.3 组件系统的组装

组件系统设计完成后,即可将各个组件按层组装,实现用户所需的各个 Web 应用。Web 应用采用两级控制策略,第一级前端控制主要进行登陆用户身份的验证和设备树请求资料的转发;第二级控制主要是验证用户的角色并根据不同的角色转发到相应的 Web 窗体页面。各个 Web 窗体页由两部分组成:视觉元素(HTML、服务器控件和静态文本)储存在 *.aspx 文件中,编程逻辑储存在 *.aspx.cs 文件中。程序集资源文件.resx 是用于编辑和定义应用程序资源的文件。在 *.aspx.cs 文件中调用表示层组件和业务逻辑层组件的服务。业务逻辑层组件

调用数据访问层的服务,通过数据访问层的服务实现和数据库的交互,实现用户的请求。系统组装如图4所示。

3 结束语

在将组件技术的思想应用到设备管理系统开发的初期,需要编写很多基本的程序组件,需要投入很大的精力

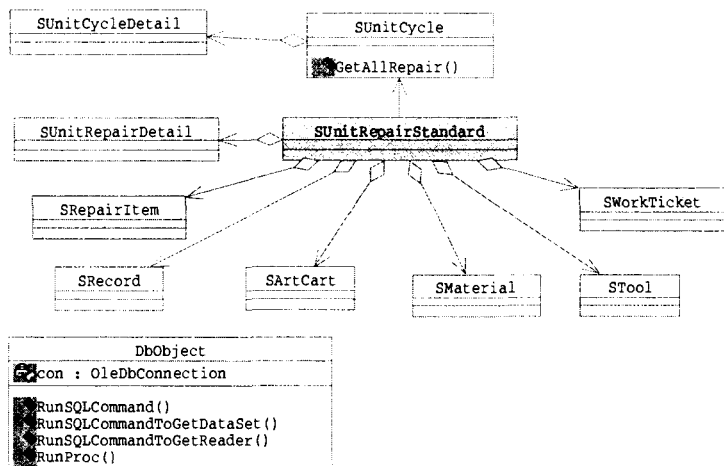


图2 机组检修标准库组件结构图

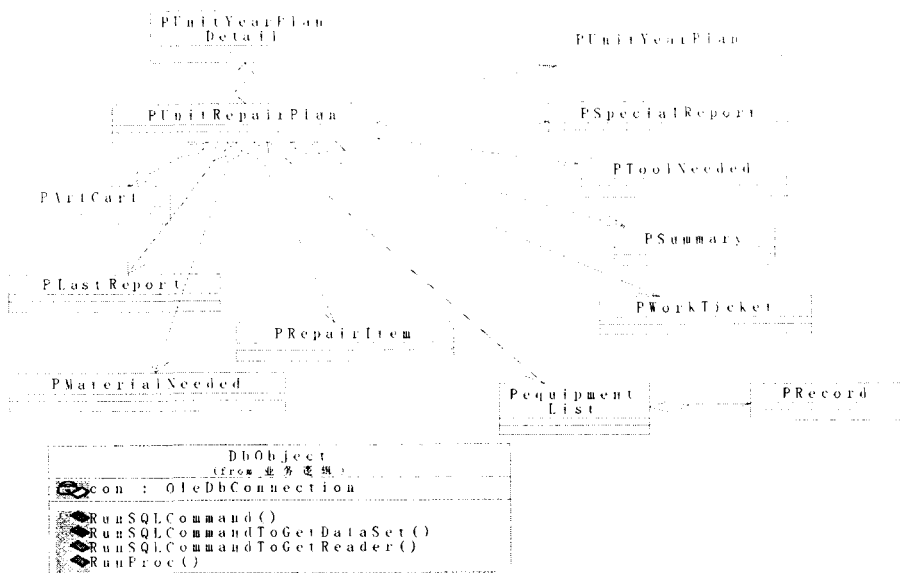


图3 机组检修计划组结构图

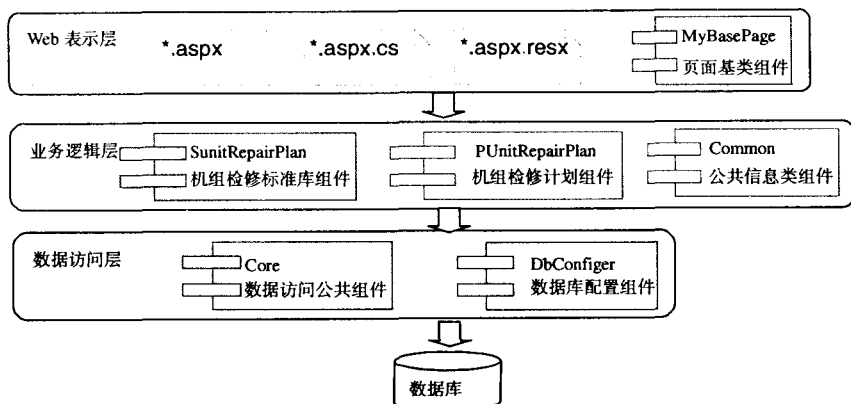


图4 系统组装图

(下转第64页)

3.2.3 员工控制器(Controllor)

控制器是 Struts 的核心,员工控制器包括员工 ActionFormBean(PersonalForm.java) 和员工 ActionBean(PersonalAction.java)两部分。其中 PersonalForm.java 这个 FormBean 中定义了员工 ActionFormBean 的属性和它所用到的 Get()与 Set()方法,而 PersonalAction.java 负责实现员工管理的映射转发。它能够针对 HTTP 页面中不同的映射请求做出相应的转发,具体包括 ADD, VIEW, EDIT, SAVE 和 REMOVE 等。

PersonalAction.java 部分代码如下:

```
//引用的类包
import javax.sql. *; import org.apache.struts.action. *;
import java.util. *; import org.apache.struts.util. *;
import java.text. *; import com. personal.model. *;
import javax.servlet.http. *; import com. personal.dao. *;
import javax.servlet. *; import com. personal.util. *;
import org.apache.commons.logging.LogFactory;
public class PersonalAction extends Action {
    public ActionForward execute(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response) {
        ActionForward myforward = null;
        String myaction = mapping.getParameter(); //取得 myaction 的值
        //根据不同的 myaction 值来执行不同的方法
        if ("".equalsIgnoreCase(myaction)) {
            myforward = mapping.findForward("failure"); //取消 Action
        } else if ("VIEW".equalsIgnoreCase(myaction)) {
            myforward = performView(mapping, form, request, response); //查看 Action
        } else if ("EDIT".equalsIgnoreCase(myaction)) {
            myforward = performEdit(mapping, form, request, response); //编辑 Action
        } else if ("ADD".equalsIgnoreCase(myaction)) {
            myforward = performAdd(mapping, form, request, response); //增加 Action
        } else if ("SAVE".equalsIgnoreCase(myaction)) {
```

```
myforward = performSave(mapping, form, request, response); //保存 Action
        } else if ("REMOVE".equalsIgnoreCase(myaction)) {
            myforward = performRemove(mapping, form, request, response); //删除 Action
        } else {
            myforward = mapping.findForward("failure");
        }
        return myforward;
    }
}
```

4 结束语

Struts 作为一种非常优秀的基于 J2EE 平台的框架正越来越多地运用到企业平台上。它最大的成功之处就在于通过 Servlet, JSP 以及标签库等一系列技术来实现 MVC 开发模式^[5]。从而有效地将系统开发过程中的表示逻辑和业务逻辑分开,使系统的设计思路更清晰,系统的修改和维护更加简单。

通过基于 Struts 框架技术的智能大厦物业管理系统的的设计,使对 JSP 页面技术、Servlet 的特性、MVC 模式以及如何将它们结合起来进行 Web 系统的开发有了很深的理解。本物业管理系统不仅能满足传统物业管理系统的要求,而且相对于一般的 Web 管理系统具有更好的可移植性和可维护性。

参考文献:

- [1] 钟 璐,潘 昊.现代软件工程[M].北京:国防工业出版社,2004.
- [2] 王肖锋,曹作良.基于 Struts 体系结构的企业级应用系统[J].天津理工学院学报,2003,19(4):63-65.
- [3] 张南平,曾昭毅,钟 璐.采用 RUP 的软件开发方法[J].微机发展,2004,14(1):81-83.
- [4] 孙卫琴.精通 Struts:基于 MVC 的 Java Web 设计与开发[M].北京:电子工业出版社,2005.
- [5] 张 杰.基于 Struts 的 Web 应用程序设计[J].现代图书情报技术,2004,107(2):33-36.

(上接第 61 页)

和时间。但是这些基本组件一旦编写完成后,可以在以后的程序开发中被频繁地重用,这样就大大提高了软件开发的效率,大幅度降低了开发费用。在系统需要修改的时候,只需修改相关的组件,而不会影响到其他部分。随着软件开发技术和方法的日益完善,使用 CBD 技术势在必行,虽然向 CBD 过渡,发展的道路并不平坦,但是 CBD 一定有广阔的应用前景。

参考文献:

- [1] Whitehead K. 基于组件开发[M]. 王海鹏,沈华峰译.北京:

机械工业出版社,2003.

- [2] 杨机智,王周敬.基于模式和组件的可重构 MIS 研究[J].合肥工业大学学报(自然科学版),2003,26(增刊):903-906.
- [3] 郭胜旺,葛 玮.构件及基于构件的开发方法研究[J].微机发展,2005,15(7):37-39.
- [4] 左爱群,黄水松.基于组件的软件开发方法研究[J].计算机应用,1998(11):4-6.
- [5] 汪 捷,杨宗凯,吴 砥.NET 技术在管理信息系统开发中的应用[J].微机发展,2004,14(11):117-120.