

配电高级应用软件的组件化研究与开发

杨秋霞¹, 李旭霞²

(1. 华北电力大学信息工程系, 北京 102206; 2. 山西省电力勘测设计院, 山西 太原 030001)

摘要: 论述了高级应用软件在配电管理系统中的地位和组件化实现对配电管理系统的重要意义, 分析了 J2EE 平台及其提供的三种重要组件技术 (EJB 组件、Servlet 组件和 JSP 组件), 介绍了 UML 统一建模语言, 研究了基于 J2EE 的高级应用软件的组件化设计与开发方法, 并以潮流计算为例具体阐述了高级应用软件的组件化实现。

关键词: 配电管理系统; 高级应用软件; 统一建模语言; J2EE 体系结构

中图分类号: TM76 **文献标识码:** A **文章编号:** 1003-4897(2005)19-0061-05

0 引言

随着城网改造的大力推进, 配电管理系统将得到快速发展。但是, 目前我国的配电管理系统还处于摸索阶段, 存在很多技术问题, 诸如网络通讯堵塞, 软件各模块可重用性不高以及模块之间的接口复杂等。组件化开发思想是将配电管理系统的各项功能采用不同的组件来实现, 一方面可以提高代码的可重用性及互操作性, 提高开发效率, 缩短开发周期, 另一方面能够提高系统的稳定性和可靠性。

高级应用软件作为配电管理系统的重要组成部分, 主要是指网络分析软件, 包括负荷预测、网络拓扑、状态估计、潮流计算、线损计算分析、电压/无功优化等^[1]。因此, 将组件化思想运用到高级应用软件开发中对于配电管理系统的发展有重要意义。

本文以 Sun 公司提供的 J2EE 作为开发平台, 采用 J2EE 平台提供的组件技术来实现配电高级应用软件的组件化开发。

1 组件化开发平台——J2EE

Sun 公司推出的 J2EE (Java 2 Enterprise Edition) 平台提供多层结构的分布式应用模型, 旨在快速设计和开发企业级应用程序。J2EE 应用程序可以是三层甚至更多层结构, 三层结构可以理解为在传统的客户机/服务器两层结构中间加了 J2EE 应用服务器层 (中间层)。根据其开发多层应用系统提供的解决方案, 应用程序的逻辑根据其要实现的不同功能, 被封装到不同的组件中组成应用程序组件, 这些组件根据其类型的不同又被部署在不同的层。采用这种根据业务分解的结果组成相应组件的方式, 大大提高了程序代码的可重用性和安全

性。同时, J2EE 平台为各组件提供了相应的容器, 容器不仅为组件提供基础设施即运行环境, 还进行组件的存活期管理、线程管理和安全性管理等, 从而简化了应用程序的建立和维护。

J2EE 平台在 J2EE 服务器层提供了三种重要的组件技术: Enterprise Java Bean (EJB) 组件、Servlet 组件和 Java Server Pages (JSP) 组件^[2]。

EJB 组件是 J2EE 平台的核心技术, 是构筑企业软件的关键, 主要包括实体 Bean、会话 Bean 和消息驱动 Bean。实体 Bean 是一种持久性的对象, 确定了数据存储的模型, 用来代表商业过程中处理的永久性数据。会话 Bean 代理客户端对服务器的请求, 实现运行于服务器上的业务逻辑。对客户端来说, 会话对象是一个非持久对象, 可以将会话对象看成是运行在服务器上的客户端程序的扩展。消息驱动 Bean 是一类特殊的 EJB, 目的在于处理 JMS (Java Message Service) 收到的消息。消息驱动 Bean 通过提供一种在 EJB 容器内部处理消息的方式来实现 JMS 的异步属性, 当应用程序客户或者一个应用程序通过 JMS 发送一条消息时, 容器调用适当的消息驱动 Bean 来处理这条消息。

Servlet 组件是一种服务器端的小程序, 允许应用程序逻辑嵌入到 HTTP 请求-响应过程中。它提供了一种扩充 Web 服务器使之能够在 HTML、XML 或者其他 Web 语言中实现动态内容的功能。

JSP 组件提供了在网页中建立组件的方式, 并且允许生成相应的网页并最终发送给客户。JSP 网页可以包含 HTML、Java 代码以及 JavaBean 组件。

2 UML 建模方法

UML (Unified Modeling Language 统一建模语

言)作为国际 OMG组织的一个标准,是一种以面向对象技术为核心的软件系统通用的可视化建模工具,用于对软件进行描述、可视化处理、构造和建立软件系统制品的文档。可视化建模用于对系统的理解、设计、浏览、配置、维护和信息控制,使用户、开发人员、分析人员、测试人员、管理人员和其它项目人员可以进行更好的沟通,更好地了解系统结构。

UML 分别从静态和动态两个角度提供了图形化的表达工具来对应用系统建模,其中使用较广泛的有用况图 (Use Case Diagram)、类图 (Class Diagram)、顺序图 (Sequence Diagram 动态)和协作图 (Cooperation Diagram)^[3]。

3 高级应用软件的组件化设计

进行高级应用软件的组件化开发,首先,我们采用 UML 从配电管理系统的整体角度出发建立组件化应用的模型,采用不同的图形化表达工具分别描述各个组件;然后,在 J2EE平台上,根据 J2EE的层次结构,采用 EJB 组件、Servlet组件和 JSP组件技术具体进行实现。下面主要以潮流计算为例介绍高级应用软件的组件化设计方法。

从统一软件开发过程^[4]的角度出发,高级应用软件建模首先要分析该软件能被用来做什么,以及谁将使用它,即进行用况分析。高级应用软件包括很多分析软件,这里只以其中的“网络拓扑”和“潮流计算”为例来说明用况模型的设计。用况模型由角色和用况组成,所有的用况必须始于角色,角色是位于所工作的系统外部的人或其它系统。

“网络拓扑”和“潮流计算”的用况模型如图 1 所示。可以看出,本系统有两个角色 (图形客户端和数据库系统)和三个用况 (潮流计算、网络拓扑和数据库访问)。图形客户端与潮流计算和网络拓扑之间的直线表示通信关系;“潮流计算”和“网络拓扑”与获取所需数据之间的带箭头的虚线表示潮流计算和网络拓扑都需要“数据库访问”这个功能,应该将这部分功能单独作为一个用况来处理,以达到组件化效果。

通过对“潮流计算”用况的具体分析,可以将潮流计算界面作为边界对象^[4],潮流计算作为控制对象^[4],潮流计算所要用到的数据作为实体对象^[4],实现潮流计算的协作图如图 2 所示。“图形客户端”通过“调用潮流用户界面”浏览界面,来调用“潮流计算”模块,“潮流计算”模块在获取了节点、母线、馈线段等实体类后进行潮流计算,最后将计算的

结果显示给客户。

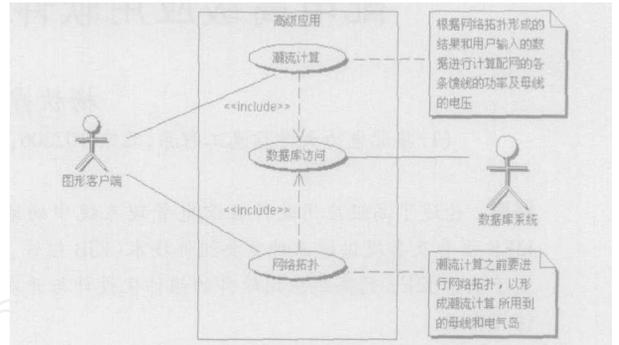


图 1 高级应用软件的用况分析

Fig 1 Use case analysis of PAS

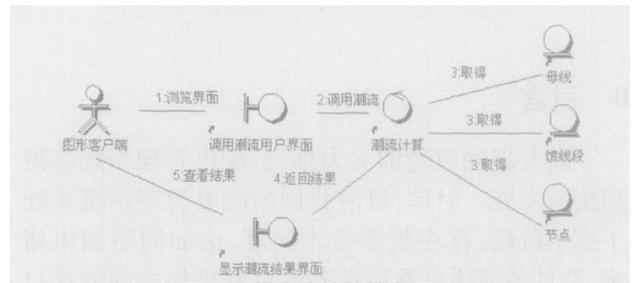


图 2 潮流计算的协作图

Fig 2 Cooperation diagram of power flow calculation

根据对高级应用软件的用况分析,现在基于 J2EE体系结构对“潮流计算”用况进行设计,其整体设计组件层次分为三层:客户层、中间层和企业数据库层,如图 3 所示。

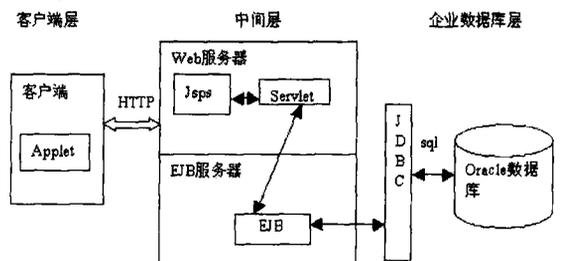


图 3 J2EE组件层次

Fig 3 J2EE component layer calculation

“图形客户端”属于客户端层,用来实现企业级应用系统的操作界面。“图形客户端”可通过将 Applet(用 Java语言编写的客户端小应用程序)嵌入到 JSP页面中实现对潮流计算的调用。

“调用潮流用户界面”、“显示潮流结果界面”和“调用潮流”属于中间层。JSP组件和 Servlet组件运行在 Web层,主要用来处理客户请求,调用相应的逻辑模块,并把结果显示返回给客户端。EJB 组件运行在 EJB层,用来实现企业级信息系统的业务逻辑。因此,“调用潮流用户界面”、“显示潮流结果

界面可以由 JSP 组件来完成,“调用潮流”可以由 Servlet 组件和 EJB 组件来完成。

“母线”、“馈线段”和“节点”等的原始数据储存于企业数据库层(本次设计采用 Oracle 数据库)。企业数据库层包括 SCADA 数据库、Net 数据库和 Topo 数据库。SCADA 数据库包括遥测数据和遥信数据。Net 数据库包括各种电气设备和用于潮流计算的数据,如节点、计算母线和电气岛等。Topo 数据库包括用于图形网络拓扑的数据。潮流计算所要用的数据,如“母线”、“馈线段”和“节点”等数据需要经 EJB 中的实体 bean (对 Oracle 数据库中原始数据的映射)来获取。

4 高级应用软件的组件化开发

从第三部分对高级应用软件进行组件化设计可以看出,组件化开发实际上是对中间层和客户端的开发,下面仍以潮流计算为例具体阐述。

4.1 中间层的组件化开发

中间层包括 Web 层和 EJB 层,因此,中间层的组件化开发就是对 Web 层和 EJB 层的开发。如第三部分所分析,本文用 Servlet 组件和 EJB 组件来完成“调用潮流”,用 JSP 组件和 Applet 完成“调用潮流用户界面”、“显示潮流结果界面”。

4.1.1 采用实体 Bean 实现对潮流算法所用到的数据库的映射

考虑到一个实体 Bean 既可以代表数据库表的一条记录,也可以维护对象的持久性,所以使用实体 Bean 实现对数据库的映射。J2EE 体系结构的 EJB 容器提供了实体 Bean 对象的对象缓冲池,客户可以多次使用缓冲池中的相同对象,也就是说少量的对象就可以供应大量用户的使用,而不需要为每个客户都生成一个实体 Bean 对象。客户通过实体 Bean 访问数据库,实体 Bean 通过 JDBC 访问数据库,这样比直接访问数据库速度要快得多,对于有大量用户访问的数据库来说,可以解决访问瓶颈问题。

数据库访问(生成 Connection 对象)属于资源占用最多的一类操作,如果每次访问数据库都要进行对象的生成,那么服务器的性能就会显著减低。JDBC 服务技术提供了连接缓冲池,连接缓冲池中预先生成多个连接(Connection)对象,客户可以多次使用同一缓冲池中的连接对象来生成或关闭一条连接(缓冲池中连接的一个句柄),也就是说缓冲池中的连接对象可以被重复使用,这样就减少了对系统资源的占用,很好地解决了服务器的性能问题。同

时这种连接缓冲池是由服务器来维护的,对开发人员是透明的。

潮流计算所用数据库中的一条记录对应一个实体 Bean,实体 Bean 的存储方式与数据库存储方式完全一样,都是永久存储方式,若实体 Bean 中的数据发生变化,那么对应的数据表的记录也发生相应变化,这种数据的同步是由 EJB 容器来管理的。实体 Bean 包括:容器管理持久性实体 Bean (CMP)和 Bean 管理持久性的实体 Bean (BMP),本文采用后者,开发者只要实现几个规定的方法即可。

4.1.2 采用无状态会话 Bean 实现潮流计算算法

会话 Bean 是运行于服务器端的业务逻辑组件,而潮流算法属于业务逻辑方法,客户在调用潮流算法时并不需要维持客户的状态,所以可采用无状态会话 Bean,无状态会话 Bean 实现起来比有状态会话 Bean 简单。潮流计算会话 Bean 中所用到的数据通过访问相应的实体 Bean 得到。

潮流计算无状态会话 bean 开发是潮流计算的核心内容,其开发过程如下:

1) 编写组成 bean 的 .java 文件:组件接口(远程接口和本地接口)、Home 接口(远程 Home 接口和本地 Home 接口)和会话 bean 类;

远程接口 (Remote Interface)

远程接口定义了 bean 类的每个商务逻辑方法,该业务方法由某个客户调用。对于在远程接口中定义的每种方法,必须在 Bean 类本身中提供一个相应的方法。

```
package Ejb Stateless Net;
public interface SweepFlow extends javax.ejb.EJBObject {
    /* 定义了计算潮流的业务逻辑方法 */
    public boolean calculateFlow (int fd_id32) throws RemoteException;
}
```

本地接口 (Local Interface)

本地客户端使用本地接口而不是远程接口来调用 bean 方法。EJB 的本地客户端指的是与所调用的远程 EJB 对象处于同一个 JVM (Java Virtual Machine) 中,如同一个 JVM 中的会话 bean 调用实体 bean,那么会话 bean 就是实体 bean 的本地客户。

```
package Ejb Stateless Net;
public interface LocalSweepFlow extends javax.ejb.EJBLocalObject {
    /* 定义了计算潮流的业务逻辑方法 */
    public boolean calculateFlow (int fd_id32);
}
```

远程 Home接口

home接口定义了远程客户能够创建和删除会话 Bean的方法。

```
package Ejb Stateless Net;
/*这是 SweepFlowBean的远程 Home接口。
* Home接口内有一个 create()方法,
*对应 SweepFlowBean内的 ejbCreate()方法。*/
public interface SweepFlowHome extends javax ejb EJB-Home {
/*该方法创建了一个会话 bean(SweepFlowBean)对象,
*必须返回该会话 bean的远程接口类型*/
public SweepFlow create() throws CreateException, RemoteException;
}
```

本地 Home接口

本地 Home接口与远程 Home接口的不同之处在于本地 Home接口可以更有效地被本地客户端使用,它定义的 create()方法返回的是会话 bean的本地接口类型。

```
package Ejb Stateless Net;
/*这是 SweepFlowBean的本地 Home接口。*/
public interface LocalSweepFlowHome extends javax ejb EJBLocalHome {
/*该方法创建了一个会话 bean(SweepFlowBean)对象
public LocalSweepFlow create() throws CreateException;
}
```

bean类

现在看一下潮流计算的会话 Bean类本身,它是会话 bean的核心,实现了潮流计算的逻辑方法,代码如下所示:

```
package Ejb Stateless Net;
//潮流计算的无状态会话 bean
public class SweepFlowBean implements javax ejb Session-Bean {
private javax ejb SessionContext context;
//会话 bean要求的方法
public void setSessionContext ( javax ejb SessionContext aContext) {
context = aContext;
}
public void ejbActivate() {
//对于无状态会话 bean,该方法保留为空
}
public void ejbPassivate() {
//对于无状态会话 bean,该方法保留为空
}
public void ejbRemove() {
```

```
//用于删除一个会话 bean对象
}
public void ejbCreate() {
}
public boolean calculateFlow (int fd_id32) {
//具体潮流计算业务方法的实现,限于篇幅省略
}
}
```

可以看出,在编写不同接口的过程中,很多代码都是容器自动生成的,这样我们就可以集中精力分析我们所要实现的业务逻辑,完成业务方法的代码编写就可以了,这就是组件带给我们的直接好处。

2) 编写部署描述符;

3) 把第 1步的 .java文件编译成 .class文件;

4) 用 jar工具创建 ejb_jar文件,包含部署描述符和 .class文件;

5) 将 ejb_jar文件部署到应用服务器上;

6) 编写一个测试客户端测试所编写的会话 bean是否正确。

会话 bean的客户端有很多种,可以是一个独立的应用程序、一个 Servlet、一个 JSP、一个 Applet 另一个会话 bean或者消息 bean和 CORBA 客户端。不管采用何种方式来访问会话 Bean,客户端都必须做如下事情:首先,查找会话 Bean的 Home接口;其次,使用 Home接口的 create()方法获得 Bean远程接口的引用;最后,通过远程接口调用 Bean类中定义的业务方法。客户并不直接调用会话 Bean类中的方法,而是通过调用在会话 Bean中透露给客户使用的 Bean类中的方法,即远程接口中定义的方法。

在后面的 4.1.3 中将以 Servlet为例说明调用潮流计算会话 bean的开发过程,在 4.3 中将说明对潮流计算的测试。

4.1.3 采用 Servlet实现对潮流计算的访问

Servlet是服务器端运行的小程序,可以对 HTTP方式的请求给予响应,利用 Servlet作为消息的生产者,客户端既可以用 Web方式实现,也可以用其它方式如 RMI/IDP方式实现。本文中的 Servlet有两种,一种是直接调用潮流计算会话 Bean,用于客户端同步调用潮流计算;另外一种作为消息 Bean的生产者,发送消息给消息 Bean,用于客户端异步调用潮流计算。下面以前一种为例说明如何调用潮流计算会话 bean。

```
package Servlet;
//调用潮流计算会话 Bean的 Servlet
//引用的包
```

```

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class SweepFlowServlet extends HttpServlet {
    //属性
    private static final String CONTENT_TYPE = "text/html";
    charset = GBK";
    Ejb Stateless Net SweepFlowHome home; //声明
    Home接口
    Ejb Stateless Net SweepFlow sweepFlow; //声明远程
    接口
    //初始化 Servlet,找到 SweepFlowHome接口
    public void init() throws ServletException {
        try {
            //获得 JND 的初始上下文环境
            javax.naming.InitialContext initial = new javax.naming.InitialContext();
            //查找会话 Bean 的 Home接口
            Object objref = initial.lookup("java:comp/env/ejb/SweepFlow");
            home = (Ejb Stateless Net SweepFlowHome) javax.naming.PortableRemoteObject.narrow(objref, Ejb Stateless Net SweepFlowHome.class);
        }
        catch (Exception e) { //捕获异常
            e.printStackTrace();
        }
    }
    //Servlet的 get方法
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        .....
        try {
            sweepFlow = home.create(); //生成远程接口的引用
            mark = sweepFlow.calculateFlow(id32); //通过远程接口调用 Bean类中的方法
            sweepFlow.remove(); //删除会话 Bean对象,容器将销毁该对象
        }
        .....
    }
}

```

4.1.4 采用 JSP显示潮流计算的结果

潮流计算的结果,即配网中各馈线的馈线段的首末端功率数据以及各母线的电压数据保存于数据库当中,客户可随时以 HTTP方式查看潮流计算的结果,查看结果以 JSP网页的形式显示。当客户端

采用同步方式调用潮流计算时,可以同步查看到潮流计算的结果。

4.2 客户端层的组件化开发

客户端层采用 Applet组件来实现,Applet嵌入到 Web层的 JSP网页中,通过 Web层的 Servlet直接或间接(通过消息 bean)访问 EJB层中实现了潮流计算的会话 Bean,实现对潮流计算的访问。

4.3 测试

本文以一个有 71 个节点、5 条馈线的配网为例,对所开发的软件进行了测试。完成一次潮流计算所需要的时间是 78 ms,能够满足现场对潮流计算速度的要求。潮流计算的结果以网页的形式显示,如图 4 所示。

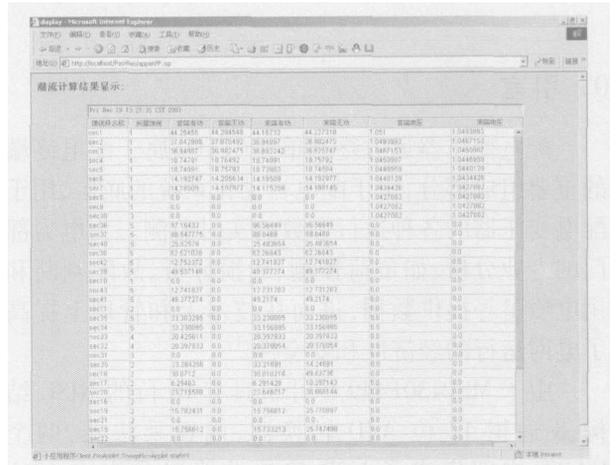


图 4 潮流计算结果显示

Fig 4 Display of power flow calculation result

5 结束语

本文采用 UML对配电高级应用软件进行建模,在建模的基础上基于 J2EE体系结构进行配网高级应用软件的组件化开发。可以看到基于 J2EE的组件化开发方法设计思路清晰,代码可重用性高,开发效率高,用该方法实现的潮流计算软件已经过调试运行良好。总之,组件化技术以及 J2EE体系结构所提供的基础运行环境对于大型的复杂的配电管理系统的软件开发具有深远意义。

参考文献:

- [1] 王士政. 电网调度自动化与配网自动化技术 [M]. 北京:中国水利水电出版社, 2003.
- WANG Shi-zheng Power Network Dispatching Automation and Distribution Automation Technology [M]. Beijing: China Water Power Press, 2003.

(下转第 74 页 continued on page 74)

- timal Power Flow: a New Decomposition Approach [A]. Proc Power Syst Computs Conf 1999. 1201-1206
- [14] Bakirtzis A G, Biskas P N. A Decentralized Solution to the DC-OPF of Interconnected Power Systems [J]. IEEE Trans on PWRs, 2003, 18 (3): 1007-1013
- [15] Irving M, Taylor G. Prospects for Grid-computing in Future Power Networks [EB/OL]. http://www.brunel.ac.uk/eesrg/research/pdf_pubs/grid03.pdf
- [16] 张伟,沈沉,卢强. 电力网格体系初探——(一)电网监控从集中计算到分布处理的发展 [J]. 电力系统自动化, 2004, 28 (22): 1-4
ZHANG Wei, SHEN Chen, LU Qiang. Framework of the Power Grid System——Part One: Development of Power

Network Monitoring from Centralized to Distributed Processing [J]. Automation of Electric Power Systems, 2004, 28 (22): 1-4

收稿日期: 2005-01-14

作者简介:

蔡广林 (1980 -), 男, 硕士研究生, 研究方向为电力系统网络计算与分布式计算; E-mail: tonsang@126.com

韦化 (1954 -), 男, 教授, 博士生导师, 主要研究方向为现代内点最优化理论及其在电力系统的应用, 涉及最优潮流、水火电力系统最优潮流、电压稳定、状态估计、网格与分布计算。

Preliminary study of grid computing and its application in power system

CAI Guang-lin, WEI Hua

(Guangxi University, Nanning 530004 China)

Abstract: This paper introduces the definition, system structure and application of other domains of grid computing. Probable applications of grid computing to power system are discussed, which mainly includes optimization of power system, power market and on-line security analysis. At last, a realization mode is predicted considering highly security requirement of power system.

Key words: grid; power system; distributing computing

(上接第 65 页 continued from page 65)

- [2] Allamaraju S, Buest C, Davies J. J2EE 编程指南 (1.3 版) [M]. 马树奇, 译. 北京: 电子工业出版社, 2002
Allamaraju S, Buest C, Davies J. Professional Java Server Programming J2EE, 1.3 Edition [M]. MA Shu-qi, Trans Beijing: Publishing House of Electronics Industry, 2002
- [3] Rumbaugh J, Jacobson I, Booch G. UML 参考手册 [M]. 姚淑珍, 唐发根, 译. 北京: 机械工业出版社, 2001.
Rumbaugh J, Jacobson I, Booch G. The Unified Modeling Language Reference Manual [M]. YAO Shu-zhen, TANG Fa-gen, Trans Beijing: China Machine Press, 2001.
- [4] Jacobson I, Booch G, Rumbaugh J. 统一软件开发过程

[M]. 周伯生, 冯学民, 樊东平, 译. 北京: 机械工业出版社, 2002

Jacobson I, Booch G, Rumbaugh J. The Unified Software Development Process [M]. ZHOU Bo-sheng, FENG Xue-min, FAN Dong-ping, Trans Beijing: China Machine Press, 2002

收稿日期: 2005-01-17; 修回日期: 2005-05-20

作者简介:

杨秋霞 (1977 -), 女, 硕士, 讲师, 研究方向为信息技术在电力系统中的应用; E-mail: yqx207@sohu.com

李旭霞 (1976 -), 女, 硕士, 助理工程师, 研究方向为配电网自动化。

Component-based research and development of PAS

YANG Qiu-xia¹, LI Xu-xia²

(1. Department of Information, North China Electric Power University, Beijing 102206, China;

2. Shanxi Electric Power Design Institute, Taiyuan 030001, China)

Abstract: Power Application Software (PAS) and its realization are very important to distribution management system (DMS). After introducing unified modeling language (UML), J2EE architecture and three important components (EJB, Servlet and JSP) provided by J2EE component-based development of PAS is studied. Taking power load flow calculation as an example, the component-based PAS realization is detailed.

Key words: DMS; PAS; UML; J2EE architecture