

JMatLink 在变压器故障诊断系统网络化迁移中的应用^①

Application of JMatLink Technology on Network Migration of Fault Diagnosis System of Transformer

罗 培 马 茜 黄辉先 (湘潭大学信息工程学院 湖南湘潭市 411105)

摘 要: 采用 JMatLink 和 ActiveX 技术解决了基于 C++ 和 Matlab 的本地变压器故障诊断系统向 Web 服务架构迁移中的资源重用问题。以 J2EE 作为开发平台,在客户端 Web 页面中嵌入 ActiveX 控件实现用户交互,在服务器端通过 JMatLink 接口对原有的变压器故障诊断系统进行 Web 集成。最后给出了一个基于 C++ 和 Matlab 的本地变压器故障诊断系统向 Web 服务架构迁移的应用实例。

关键词: Web Services 故障诊断 ActiveX JMatLink

1 引言

随着电力信息化建设的不断深入,为了解决电力企业计算环境日趋复杂的分布异构问题,基于 Web Services 的电力企业应用集成 EP-EAI(Electric Power enterprise-Enterprise Application Integration)得到了研究人员的极大关注。由于采用了 SOAP、XML、UDDI 等标准化技术,使得基于 Web Services 的 EP-EAI 具有很好的跨平台和语言的能力,能够将分布于不同平台的异构系统以松耦合的方式进行集成,可以很好地满足电力企业分布异构应用集成的要求^[1-4]。

基于 Web Services 的 EP-EAI 研究难点之一是:电力企业在发展过程中已陆续应用了一定规模的信息系统和应用模块,比如电力系统状态监测和故障诊断系统,如何可靠而方便地将这些资源集成进 Web 环境,而不必做大的修改或重新开发。通过对已建立的信息系统和应用模块进行网络化迁移,可以重用现有的资源,是一种高效的集成方法。

目前常用的方法一般是在服务器端以 Socket 方式进行通讯,客户端采用 Java Applet 或 ActiveX^[5],编程工作量大,系统之间紧密耦合,互操作性差。本文针对基于 C++ 和 Matlab 的变压器故障诊断系统,在服务器端用 JMatLink(Java 与 Matlab 接口)实现 Java 代码

与 Matlab 之间的接口,在客户端浏览器中嵌入 ActiveX 控件实现与用户交互,这样可以尽量重用原有代码,提高开发效率。因此,采用 JMatLink 和 ActiveX 技术对基于 Matlab 的变压器故障诊断系统进行网络化迁移是很合适地。

本文用基于神经网络的变压器故障诊断系统作为研究对象,对基于 Matlab 的变压器故障诊断系统进行网络化迁移,实现了基于 Web Services 的变压器故障诊断系统设计。

2 系统框架结构

系统框架结构如图 1 所示。在服务器端利用 J2EE 和 WebSphere 发布服务,客户端通过浏览器访问包含 ActiveX 控件的 Web 页面,调用 Web 服务及封装系统功能。

EJB 容器为服务器端容器,包含的组件为 EJB,主要用于服务器端功能逻辑的实现。

现有的变压器故障诊断系统采用 C++ 和 Matlab 设计,C++ 编写用户交互界面,Matlab 编写故障诊断程序。为了尽可能重用现有资源,利用 JMatLink 保留 Matlab 故障诊断程序代码,将接口程序用 Java 改写,

① 基金项目 湖南省自然科学基金资助项目(06JJ5112)

这样大大减轻了开发强度,提高了代码质量。

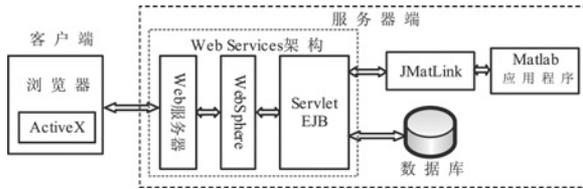


图 1 系统框架结构图

3 系统设计的关键技术

3.1 客户端 ActiveX 控件的包装

通过包装原来的故障诊断系统的方法建立客户端的 ActiveX 控件,可以尽可能多的重用原有代码。在 VC++ 环境下,通过使用 ActiveForm 窗体设计器来包装 ActiveX 控件,用户可以将已有的各种控件集成到界面上,绘制各种监控画面、图表和曲线。需要包装的原控制系统工程名为 :OriControl,包装后的控件名为 NewOcx,步骤如下:

(1) 建立新控件工程 NewOcx:使用 MFCActiveX ControlWizard 建立控件工程 NewOcx;

(2) 引入资源:将原程序工程 OriControl 中的工具栏、菜单、鼠标指针、对话框、快捷键、位图、图标、字符串表等资源添加到控件工程中;

(3) 引入原程序的类:在 OriControl 中选择需要包装到控件里的功能模块用到的类及框架类(COriControlFrame)、文档类(COriControlDoc)和视图类(COriControlView),添加到控件工程中;修改控件的 CNewOcxCtrl 类定义:增加一个控件 CNewOcxApp 类的公共成员变量指针:public :CNewOcxApp m_pApp;修改控件 CNewOcxCtrl 类中响应 WM_CREATE 消息的 OnCreate 函数;

(4) 修改 CNewOcxApp :InitInstance() 的实现,添加如下语句:

```
if ( !bInit )
{
    CMultiDocTemplate * pDocTemplate ;
    pDocTemplate = new CMultiDocTemplate (
        IDR_MAINFRAME ,RUNTIME_CLASS( COriControlDoc ),
        RUNTIME_CLASS( CChildFrame ),
        RUNTIME_CLASS( COriControlView ) );
```

```
AddDocTemplate( pDocTemplate );
CMainFrame * pMainFrame = new CMainFrame ;
if( ! pMainFrame -> LoadFrame( IDR_MAINFRAME ))
    return FALSE ;
m_pMainWnd = pMainFrame ;
}
```

(5) 添加消息映射:将和需要包装的功能模块有关的消息映射宏及相应的消息响应函数添加到控件中。另外,由于控件的消息映射机制和普通应用程序有所不同,它不会自动执行 ON_UPDATE_COMMAND_UI 消息响应的函数,需要手动添加工具栏的 OnUpdateCmdUI 函数。

经过上述步骤,即可将一个原来用 VC++ 编写的 MD 风格的应用程序包装成 ActiveX 控件。其他风格的界面,也可以采用类似的方法。

3.2 JMatLink 的实现

JMatLink^[6,7]是由德国 Stefan Muller 开发,用于在 Java 语言与 Matlab 之间建立接口。使用 JMatLink 可以实现 Java 调用 Matlab 的引擎函数。首先用 C/C++ 代码把 Matlab 引擎库封装到一个共享库中,然后用 JNI(Java 本地平台接口)接口调用共享库,最后在 Java 中对其应用并实例化,从而实现对 Matlab 引擎的调用,完成数值计算、分析和图像显示等功能。

根据 JNI 程序得编写规范,接口实现过程步骤如下:

(1) 编写调用共享库的 Java 类。

在 java 程序中,首先需要在类中声明所调用的库的名字:

```
Static{
    Try { System. loadLibrary( "JMatLink" );}
    Catch( UnsatisfiedLinkError e )
        { System. out. println( "ERROR :Could not load the JMatLink library" );}
}
```

(2) 编译 JMatLink. java 得到 JMatLink. class。

(3) 使用 javah -jni 命令产生 JMatLink. h 文件

(4) 编写本地化方法的 C/C++ 实现,即编写 JMatLink. c 文件。

JNI 对 Java 和 C/C++ 的数据类型进行了映射,jint 是以 JNI 为中介使 Java 的 int 类型与本地的 int 沟通的一种类型。本地方法的第一个参数是 JNI 接口指

针,接口指针指向一个指针数组,数组的每个元素指向一个接口函数,本地代码通过调用 JNI 接口函数访问 Java 虚拟机的特定功能。第二个参数取决于此方法是静态还是非静态。非静态本地方法的第二个参数是调用本地方法 Java 类所属的对象,静态方法的第二个参数则是调用本地方法的 Java 类。剩下的参数和通常的 Java 法参数一致。需要说明的是,C 和 C++ 的代码几乎相同,唯一的差异在于用来访问 JNI 函数的方法。在 C 中,JNI 函数使用“(*env) ->”作前缀,目的是为了取出函数指针所引用的值;在 C++ 中“env ->”类拥有处理指针查找的内联成员函数。

(5) 引入 MATLAB 的函数库,把 JMatLink.c 编译成共享库文件 MatLink.dll。

JMatLink 接口通信机制如图 2 所示。

Java engine 将组件信息和待处理的数据通过 JNI 传入 Matlab 计算引擎做相应地处理,其结果再被传给客户端,由此,通过 JMatLink 接口实现了 Java 语言与 Matlab 之间的函数调用,完成了对原诊断系统的 Java 化改造。由于 JNI 是 Java 直接通过一个本地接口调用本地的功能库,相当于进程间的交互,所以相对于 COBRA、Socket 和 RMI 而言,JNI 速度最快,因此,采用 JNI 可以满足系统实时性要求。

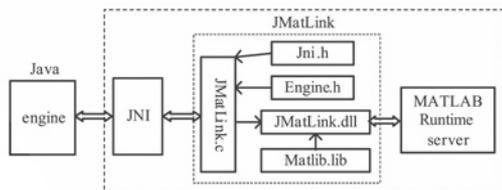


图 2 JMatLink 通信机制图

4 应用实例

目前常用的 Web Services 开发框架有 J2EE 平台和 .NET 平台, .NET 平台用户界面友好,内嵌对 SOAP 和 XML 的支持,开发效率高,具有 Windows 平台下的语言无关性。而 J2EE 平台提供了更好的开放性和安全性,并可用 JNI 实现对原有代码的重用。考虑到网络平台的环境比较复杂以及以后该课题的扩展性,所以本系统采用 J2EE 作为开发平台。

图 3 给出了基于 C++ 和 Matlab 的本地变压器故障诊断系统向 Web 服务架构迁移的应用实例。

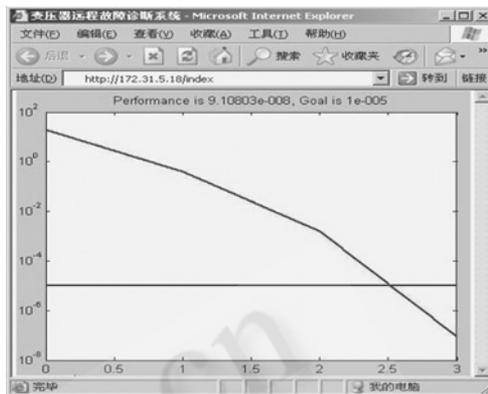


图 3 基于 Web Services 的变压器故障诊断系统界面

5 结束语

本系统采用 JMatLink 和 ActiveX 技术实现了基于 Matlab 的变压器故障诊断系统的网络化迁移,具有开发方便、效率高;开发过程简化、结果精确;用户界面友好等优点。由于采用 Web Services 作为解决方案,该系统还具有好的扩展性和重用性,可以与电力系统的远程状态监控相结合,进一步扩展其应用范围。

参考文献

- 1 麻丽娜,苑津莎,李新叶.基于 Web Services 的电力企业应用集成技术研究与实现[J].电力系统通讯,2006,26(157):40-42.
- 2 方烁,梁成辉,徐庆平,云昌钦.IEC 61970 标准中 CIS 的 Web 服务定义与实现[J].电力系统自动化,2006,30(15):81-84.
- 3 陈传波,张道杰,李涛.基于 Web 服务的企业应用集成模型研究[J].计算机工程与科学,2004,26(12):15-19.
- 4 李华,莫娟.基于 Web 模式的变压器绝缘故障诊断系统[J].高电压技术,2005,31(6):12-14.
- 5 刘锋光,王万良,蒋一波,毛明杰.基于 JSP/JavaBean 和 ActiveX 技术的远程监控[J].计算机工程与应用,2006,(3):197-199.
- 6 JMatLink [EB/OL]. www. held - mueller. de/ JMatLin/,2006.
- 7 郝雪峰,须德,范莹. JSP 环境下基于 Matlab 引擎的数据可视化方法[J].计算机工程与应用,2004,(22):110-112.
- 8 陈维荣,宋永华,孙锦鑫.电力系统设备状态监测的概念及现状[J].电网技术,2000,24(11):12-17.