

NDT650 综合自动化系统支撑软件架构

貌学良

(南京信息职业技术学院, 江苏 南京 210046)

摘要: 在简单分析电力系统仿真软件的基础上, 论述开发 NDT650 综合自动化系统支撑软件的必要性, 介绍了采用面向对象编程语言 Visual C++、组件、动态连接库、数据访问对象、数据库技术、Windows Socket 套接字技术进行软件架构的方法, 该软件已通过鉴定, 并成功应用于多个变电所。

关键词: 界面; 通信; 数据库; 模块; 装置

Frame of support software used for NDT 650 integrated automation system

MAO Xue-liang

(Nanjing College of Information Technology, Nanjing 210046, China)

Abstract: On the basis of simple analysis of simulation software for electric power system, the development necessity of the support software used for NDT 650 integrated automation system is discussed. The method of the software development is introduced using object oriented programme language Visual C++, component, Dynamic-Link Library, Data Access Objects, database and Windows Socket technology. This software has been identified and used in many substation successfully.

Key words: interface; communication; database; module; equipment

中图分类号: TM76 文献标识码: A 文章编号: 1003-4897(2007)03-0051-04

0 引言

目前, 许多电力系统继电保护工作者, 在进行理论研究、电路设计、算法探讨、模型建立时, 往往借助于仿真软件^[1~3]。然而, 仿真软件在提高研制新装置效率的同时, 也存在了一些局限性。保护装置在出厂前, 都必须进行定值整定、参数设置, 如果在装置上直接输入, 往往比较麻烦, 另外, 只有进行过保护功能测试的装置, 在现场动作才可靠。因此, 为了提高保护质量和工程的效率, 特研制 NDT650 综合自动化系统支撑软件。该软件是整个 NDT650 综合自动化系统的一个组成部分, 配合保护测控系统工作。

该软件以异构系统 Windows 和嵌入式 nucleus 组成操作系统平台, 采用面向对象的 VC++ 6.0 编程语言, 组件技术、动态连接库 (Dynamic-Link Library) 技术、DAO (Data Access Objects) 策略以及套接字技术开发。组件的采用, 克服了不同通信协议以及异构操作系统之间的不兼容问题, 从而实现跨平台的可操作性^[4]。整个支撑系统软件由

五个分析软件组成。

1 支撑系统软件模块划分

NDT650 支撑系统软件根据开发、分析与维护的对象的不同, 分为微机馈线保护测控维护分析软件 (以下简称分析软件)、变压器分析软件、变压器后备分析软件、电容器分析软件、备自投分析软件, 而每个分析软件又由不同的功能模块组合。如图 1 支撑系统软件模块, 其中, 通信监视、内存读写、时间校正、定值管理几个模块, 通过动态连接库实现代码共享与复用。为此, 各分析软件的应用类中嵌入以下动态连接库:

```
#pragma comment(lib, "communicate.lib") //通信
动态连接库
#pragma comment(lib, "value.lib") //定值管
理动态连接库
#pragma comment(lib, "watch.lib") //通信监
视动态连接库
#pragma comment(lib, "memoryrw.lib") //内存读写
动态连接库
```

```
#pragma comment(lib, "timeset.lib") //时间校正动态连接库
```

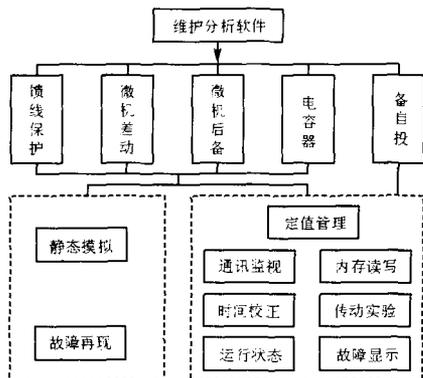


图1 支撑系统软件模块

Fig.1 Module of support software

2 系统软件架构

2.1 支撑系统软件结构

NDT650 支撑系统软件包括五个相对独立的分析软件，而每个分析软件又包括众多的功能模块，所以，借助文档模板，采用多文档、多窗口思想研制开发。文档模板管理一组特殊的文档、文档边框、视图，将三者有机地结合为一个整体，其中，文档类型决定了文档边框类型和视图类型。因此，根据各功能模块的不同，分别创建不同的文档模板。例如，馈线分析软件，除了应用框架创建的默认文档模板对象之外，又另外创建了内存读写、传动实验、静态模拟等文档模板对象。

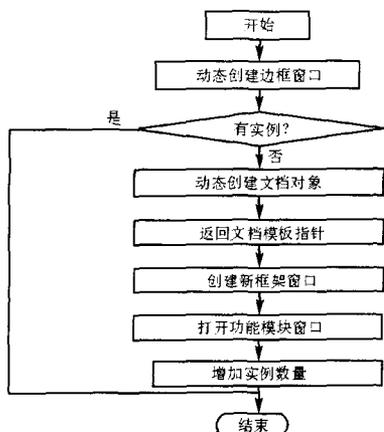


图2 模块窗口创建

Fig.2 Establishing of module window

链表中。每当打开不同类型的文档时，应用对象必须获得相应文档模板类对象，并根据该对象创建相应的文档边框窗口（MDI 子窗口）、视图类窗口，以显示文档。文档模板类对象可通过应用类的 Get First Doc Template Position() 和 Get Next Doc Template() 获得。由于牵涉链表操作，消耗的系统资源相对较大，且支撑软件文档模板类对象数量不多，所以，不采用链表，而直接通过堆中的文档模板类对象指针来访问对象。功能模块窗口的创建过程，见图2 模块窗口创建。

支撑系统软件根据需要，可同时打开多个功能窗口，但为便于管理，在同一时刻每个类型的功能模块窗口只能有一个实例。各功能模块工作过程：
 (1) 判断触发功能模块类型，并以此设置类型标识码。
 (2) 获取功能模块中通信发送线程和接收线程指针。
 (3) 如果是启动功能模块，首先，给发送线程和接收线程互斥信号量加锁，此时，如果还没有打开串口，则打开串口，设置打开标志。其次，根据类型标识码，决定是否需要向保护装置发送命令以及命令的类型，最后，解锁发送线程和接收线程互斥信号量。如果是关闭功能窗口，第一步，清除接收线程的打开标志，第二步，给发送线程和接收线程互斥信号量加锁，第三步，返回接收线程退出码，当该线程还处于活动状态时，则结束线程，并释放线程句柄，最后，解锁发送线程和接收线程互斥信号量。见图3 功能模块工作流程。

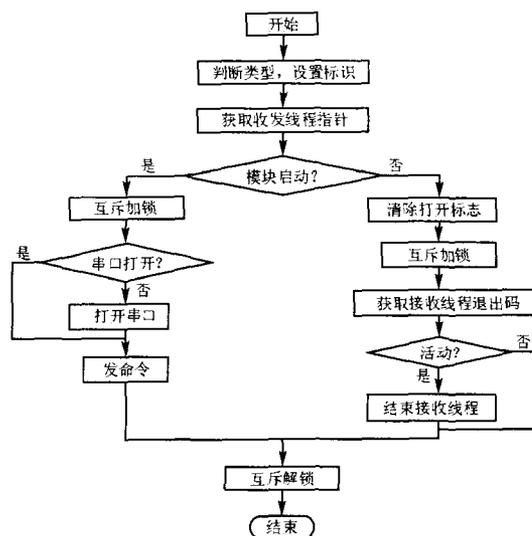


图3 功能模块工作流程

Fig.3 Workflow of function module

通常，应用程序根据不同的文档类、文档边框窗口类和视图类，分别动态地创建文档模板类对象，且将返回的对象指针加入文档模板类对象指针

2.2 支撑系统软件界面

在电力系统分析时,为便于对比,往往从几个方面同时进行,需要同时启动多个功能模块,因此,支撑系统软件采用多窗口、多线程的运行机制。其界面主要包括窗口、功能模块驱动菜单、通信热态工具条、指导性状态条、操作板,其中,驱动菜单用于打开各功能模块;通信热态工具条由与菜单具有相同功能的图形化按钮、提示通信状态的串口连接控件窗和参数控件窗组成;而指导性状态条则由触发操作板的图形化按钮、提示信息、进度条和时钟框组成。以下着重介绍窗口、操作板。

2.2.1 窗口界面

为便于对电力保护装置进行全方位的分析,支撑软件除了采用多窗口方式外,还采用了切分窗口技术;另外,彩色按钮、模拟操作声响和滚动提示文字的使用,增强用户的友善性。

1) 静态切分窗口,分块显示保护信息

在故障录波分析模块中,由于待分析的录波报文中的信息种类、数量固定,因而,采用窗口静态切分技术。首先,将框架窗口用户区分成三行一列三个窗格,第一个窗格包含故障录波波形分析视图类,第二个窗格包含保护状态分析视图类,第三个窗格将窗口作为容器处理。其创建过程如下:

```

BOOL CMychildwnd::OnCreateClient(LPCREATESTRUCT lpcs,
CContext* pContext)
{ if (!m_wndSplitter.CreateStatic(this, 3, 1)) //
创建三个窗格
    {出错误处理 }
...
if (!m_wndSplitter.CreateView(0, 0, RUNTIME_CLASS
(CCurveView), CSize(rect.Width(), rect.bottom/2-20),
pContext)) //创建包含故障录波波形分析视图类的窗格
    {出错误处理 }
if (!m_wndSplitter.CreateView(1, 0, RUNTIME_CLASS(
CStateView), CSize(rect.Width(), 100), pContext))
//创建包含状态分析视图类的窗格
    {出错误处理 }
if (!m_wndSplitter.CreateView(2, 0, RUNTIME_
CLASS(CPaneFrame),
CSize(1, 1), pContext)) //创建包含容器窗口类的窗
格
    {出错误处理 }
p_curveview=(CCurveView*)m_wndSplitter.GetPane(
0, 0); //返回录波波形分析视图类窗口指针
p_stateview=(CStateView*)m_wndSplitter.GetPane

```

```

(1, 0); //返回状态分析视图类窗口指针
p_paneview = (CPaneFrame*)m_wndSplitter.
GetPane(2, 0); //返回容器窗口指针
return TRUE;
}

```

其次,再将第三个容器窗格分为两个子窗格,一个包含保护出口参数显示类 CReportView,另一个包含保护动作类型分析类 Cprotectstave。经过这样切分,程序在运行过程中可动态改变窗格大小,灵活分析故障录波中的不同部分。

2) 运行界面模式的设置

为达到提高界面的友善性的目的,在应用程序类中,设置各功能模块都能共享的数据成员 m_ColButton、m_Animal、m_Sound、WndNum 分别控制按钮、滚动文字、模拟声响属性和同时打开窗口的数量,其中, m_ColButton 为真,则采用彩色按钮, m_Sound 为真,则增加模拟声响, m_Animal 为真,用滚动文字显示提示信息。彩色按钮创建过程如下:

```

void COuttestView::OnInitialUpdate()
{ CFormView::OnInitialUpdate();
...
if(m_pApp->m_ColButton)
    {p_button=(CButton*)GetDlgItem(IDC_BUTTON1);//
返回按钮对象指针
        style=p_button->GetButtonStyle();//返回
按钮风格
        style|=BS_OWNERDRAW; //在原按钮风格基础
上,增加自画功能
        p_button->SetButtonStyle(style); //设置按
钮新风格
        m_button1.Setbt(20, 255, 0, FALSE); //设置背
景色
        m_button1.SubclassDlgItem( IDC_BUTTON1,
this );//子类化按钮
        ...
    }
UpdateData(false);
}

```

2.2.2 集成操作面板

操作面板模仿家用电气的遥控器,集成几乎所有的操作功能,操作非常方便。其实现策略为:首先创建操作面板资源,并使其中的按钮标识分别与工具条中的按钮标识相对应,其次,在指导性状态

栏上创建位图按钮对象，并为其装载正常和按下两种位图资源，最后，将操作面板与状态栏中的触发按钮关联。

2.3 数据库构成

通常，保护测控装置都必须根据定值来工作，定值的整定至关重要。采用何种手段进行定值管理，才能提高保护工作效率是必须考虑的问题。定值管理的方法很多，支撑系统软件采取的策略是：数据库系统结合定值模块进行定值管理。

2.3.1 定值数据库的创建及优化

首先，依据保护测控装置的类型，分别创建不同的 Access 数据库，例如馈线保护数据库 fe_fixvalue，主变保护数据库 su_fixvalue。其次，为各数据库创建 8 个定值表，并以“区号 0”到“区号 7”作为表名，用于存储 8 套定值。由于“定值”字段存储的定值类型各异，有整型、字符串型、浮点型，这给设定字段类型带来不便，如果设为整型则无法满足字符串型、浮点型，如果设为浮点型又无法满足整型和字符串型，而且浮点型定值在存储显示时易造成精度损失。为解决这一问题，支撑软件将“定值”字段处理为字符串型，并在应用程序初始化文件中，为每一定值定义一个小数点位置和单位数据项，其中第一位含义如表 1。最后，编辑定值，并保存数据库。程序执行时，根据第一位含义，将字段内容处理为相应格式。

表 1 数据项首位含义

Tab.1 Meaning of first data item

序号	数据项首位	含义
1	0	整数
2	1	1 位小数
3	2	2 位小数
4	3	3 位小数
5	4	十六进制值
6	5	控制字 1
7	6	控制字 2

2.3.2 通过定值管理模块编辑数据库中定值

在定值管理模块中，采用数据访问对象 DAO (Data Access Objects) 策略，通过 Microsoft Jet 数据库引擎对 Access 数据库进行访问，实现定值的整定。

为实现代码的重用与共享，支撑软件将定值管理模块封装于 value.dll 动态连接库中。当模块编辑数据库时，根据调用 value.dll 的应用程序类型

m_DeviceType 构建与之相对应的数据库文件全路径，例如主变维护软件则构建主变保护数据库 su_fixvalue 全路径。紧接着动态创建数据库 CdaoDatabase 类对象、表存储定义 CdaoTableDef 类对象，并分别打开数据库及其中选择的表，而编辑任务则由派生于 CdaoRecordset 的记录集类 Cfixvalue 完成。Access 数据库定值表中的字段与定值管理模块 Cfixvalue 类数据成员的对应关系如下：

```
void
Cfixvalue::DoFieldExchange(CDaoFieldExchange* pFX)
//{{AFX_FIELD_MAP(Cfixvalue)
    pFX->SetFieldType(CDaoFieldExchange::outputColumn);
    DFX_Long(pFX, _T("[序号]"), m_Serial);
    DFX_Text(pFX, _T("[定值符号]"), m_Symbol);
    DFX_Text(pFX, _T("[定值名称]"), m_Name);
    DFX_Text(pFX, _T("[建议范围]"), m_Range);
    DFX_Text(pFX, _T("[定值]"), m_Set);
    DFX_Text(pFX, _T("[单位]"), m_Unit);
//}}AFX_FIELD_MAP
}
```

2.4 数据通信

数据通信的方式很多，其中，Windows Socket 套接字技术^[6]在网络通信中应用非常广泛。按照这种方式，服务器方需创建两个套节字，既监听套节字和读取套节字，客户端需创建一个套节字，通常需地址绑定。基于这种套接字技术的通信非常可靠，所以，为提高与保护装置的通信效率，创造性地将套接字通信思想应用于电力保护装置的串口通信中。

基于套接字技术的串行通信，通过 RS232 串口连接运行支撑软件的计算机（以下简称支撑机）与保护、测控装置，构成客户/服务器运行模式。支撑机通过串口向装置发命令或写诸如保护定值等数据。支撑软件所有的通信功能都由封装了通信类 CMycomm 的 communicate.dll 动态链接库实现，通信过程为：

1) 实例化通信类 Cmycomm。

2) 通过调用类 Cmycomm 成员函数 OpenCOM 打开串口。

```
BOOL CMycomm::OpenCom(int port, int baud,
int parity, int databits, int stopbits, int
timeout, CFileException *e)
```

```

{ CString portNo;
portNo.Format("COM%d:",port); //根据指定的port
定义串口
if (Open( portNo, CFile::modeReadWrite e)) //打
开串口设备文件
{DCB dcb; //串口通信设备控制设置结构
::GetCommState( (HANDLE)m_hFile, &dcb); //获取
串口通信设备控制值
if(baud!=-1) dcb.BaudRate = baud; //配置
波特率
if(databits!=-1) dcb.ByteSize = databits; //配
置数据位长度
if(stopbits!=-1) dcb.StopBits = stopbits; //配
置停止位
if (parity!=-1) dcb.Parity = parity; //配
置奇偶校检
::SetCommState( (HANDLE)m_hFile, &dcb); //设置
新的串口通信设备控制值
COMMTIMEOUTS cto; // 定义通信延时结构
::GetCommTimeouts( (HANDLE)m_hFile, &cto); // 获
取通信延时
配置通信延时参数
::SetCommTimeouts( (HANDLE)m_hFile, &cto); //
设置新的通信延时
return(TRUE);
}
return(FALSE);
}

```

3) 填写如下形式的通信帧信息。

功能码
帧数据长度
帧数据
循环校验码

4) 启动发送线程, 通过调用类 Cmycomm 成员函数 Send 发送信息。

5) 启动接收线程, 通过调用类 Cmycomm 成员函数 Listen 监视串口。

6) 串口一旦收到信息, 通过报文队列管理类 CwzdQueue 将报文写入报文队列中, 同时向相应的窗口发有数据达到的消息。

7) 窗口收到消息后, 通过 CrepQueue 类的 RepGet 函数将报文从队列中取出并加以处理。

3 结语

NDT650 支撑系统软件已研制成功, 并已通过国家电力公司鉴定。该软件已在 NDT650 综合自动化装置的研制过程中得到应用: 监控板或保护板中的问题, 采用常规的方法很难测出, 而通过内存读写模块, 向保护测控装置指定单元写入特定数据, 在其运行之后, 再分析从内存读取的数据, 可较为容易地诊断出问题所在; 通过分析通信监视模块所收的报文, 可以判断装置动作是否符合预定的逻辑。以上两模块加速了 NDT650 综合自动化馈线、主变等装置的开发进度。

在贵州遵义、太白变电所以及朔黄铁路沿线各变电所、开闭所实际工程中, 用到几十个装置, 如果按常规, 一个一个手工输入保护定值、参数、时间, 工作量巨大。当时为提高工作效率, 采用定值管理模块, 通过串口成批输入定值, 大大地减少了整定时间; 借助时间校正模块校正装置时间, 使所有的装置具有统一的时间; 为验证工程中的保护装置在出现“故障”条件时, 动作的可靠性、灵敏性、及时性以及选择性, 采用了故障再现和静态模拟两模块, 从动、静两个角度分别进行: 首先, 在静态模拟模块中, 将静态模拟的故障波形数据, 通过 RS232 串口下装到保护装置, 模拟一次设备故障, 并根据保护装置动作情况, 判断保护是否有问题。例如: 馈线速断保护电流定值为 8 A, 速断保护出口时间为 100 ms, 过流保护电流定值为 4 A, 过流保护出口时间为 500 ms, 为检验速断保护, 在输入值略大于 8 A、周波数大于 5 周波 (每周波 20 ms, 5 周波即 100 ms) 的基波电流情况下, 如果装置不动作, 说明速断保护有问题; 在输入值大于 4 A、周波数大于 25 周波的基波电流时, 如果过流没有出口, 说明装置过流保护部分有问题。其次, 在故障再现模块中, 将原先故障时所录的故障录波数据下装到保护装置, 从而实现故障过程的再现, 根据装置动作情况判断其工作是否正常; 另外, 每个装置出口通道都经过传动实验检验, 以确保故障时各出口通道畅通。在上述工程中, 采用故障再现、静态模拟和传动实验检验出一些有问题的装置, 并及时进行了调整, 而通过检验的装置在现场运行良好, 取得良好的社会效益和经济效益, 这充分证明 NDT650 支撑系统软件具有相当的实用价值。

(下转第 76 页 continued on page 76)

相保护动作,三跳本开关,故障快速切除。

3) 手动分闸,开关非全相。此时,开关非全相保护动作,三跳本开关,若不成功,而开关失灵保护没有启动元件不能动作,故障不能切除。

而且在第二种情况下,仅靠启动开关失灵保护不一定能完成动作。因为这时发电机的负荷电流非常小,不足以启动失灵保护的电流判据。所以需要为发电机开关的非全相保护特别设计“失灵逻辑”。在三跳本开关不成功时,采用开关辅节点非全相逻辑加负序电流元件或零序电流元件(一般开关保护只采用过流元件,灵敏度不够)把关,跳开相邻元件。在时间设定上,由于发电机开关没有重合闸问题,其动作延时可以相对缩短一些,但不建议非全相 0 s 跳闸,应至少延时一个 Δt 。

5 关于重合闸

华东电网 500 kV 系统线路均采用单相重合闸方式。处于系统联络变电站地位的发电厂,出线重合闸是肯定要投用的,只是需要采用系统侧先合的原则;单线送单变的接线方式,为了减少发电机非全相运行时间,电厂侧可以不用重合闸,这样电厂侧开关可以选用三相操作机构,利于电厂运行。但是若出线较长,处于雷区,单相瞬时故障概率较高,电厂应综合考虑非全相运行和机组停运之间的利弊而决定是否启动线路重合闸。系统侧一般启用重合闸,便于电厂事故后恢复。

6 结论和建议

(上接第 55 页 continued from page 55)

参考文献

- [1] 张福生,吴秋瑞,张建成.电力系统微机保护的 MATLAB 仿真插值算法[J].微计算机信息,2006,2(1).
ZHANG Fu-sheng,WU Qiu-rui,ZHANG Jian-cheng.The Interpolation Algorithm and Its MATLAB Emulation on Electric Power System's Microcomputer Protection[J].Microcomputer Information,2006,2(1).
- [2] 施文济.如何实现 MATLAB 与 SPICE 仿真分析之间的转换[J].微计算机信息,2006,2(1).
SHI Wen-ji.How to Realize Transformation Between SPICE and MATLAB in Simulation Analysis[J].Microcomputer Information,2006,2(1).
- [3] 高松,贺仁睦,马进,等.电力系统动态仿真误差评定准则研究[J].电力系统自动化,2006,30(4):6-10.
GAO Song, HE Ren-mu, MA Jin, et al. Error Criteria on Power System Dynamic Simulation Validation[J]. Automation of Electric Power Systems, 2006,30(4):6-10.

继电保护是电力系统稳定第一道防线的重要手段,而并网大机组是电力系统的重要组成部分,如何保证大机组在系统发生故障和其他异常工况下的安全稳定运行,是关系到电力系统稳定的重大问题,也是保证机组安全的大事。所以,不论体制如何变化,电力系统在稳定运行方面的一体化问题是不会改变的,因此应重视并网大机组和输电网系统在保护分界面上的协调配合问题,并制定有关工作方案。要做好这项工作,不仅要从保护的整定运行方面,还需要在保护原理选择,逻辑设计和设备选型多方面综合考虑,必要时要对主设备运行参数提出要求。

参考文献

- [1] 王维俭.大型机组继电保护理论基础[M].北京:中国电力出版社,1997.
WANG Wei-jian.The Foundational Theory of Generator-transformer Unit Relay Protection[M]. Beijing:China Electric Power Press,1997.
- [2] 王梅义.电网继电保护应用[M].北京:中国电力出版社,2000.
WANG Mei-yi.The Application of Grid Relay Protection[M].Beijing:China Electric Power Press,2000.
- [3] 华东电网 2006 年年度运行方式[Z].2006.
The Annual System Analysis of East China Grid[Z]. 2006.

收稿日期:2006-11-04; 修回日期:2006-11-21

作者简介:

聂宇本(1969-),女,长期从事电力系统专业工作;
E-mail:nie-yb@ec.sp.com.cn

唐渝南(1969-),女,高级工程师,长期从事继电保护专业工作。

- [4] 王映辉,冯德明.大规模软件构架技术[M].北京:科学出版社,2003.
WANG Ying-hui,FENG De-ming.The Truss Technology for Large Scale Software[M].Beijing:Science Press,2003.
- [5] 官章全,刘加明.Visual C++ 6.0 类库大全[M].北京:电子工业出版社,1999.
GUAN Zhang-quan,LIU Jia-ming.Collected Edition of Visual C++ 6.0 MFC Library[M].Beijing:Publishing House of Electronics Industry,1999.
- [6] Swanke J E.Visual C++扩展编程实例[M].北京:机械工业出版社,2000.
Swanke J E.VC++ MFC Extensions by Example[M]. Beijing:China Machine Press,2000.

收稿日期:2006-06-24; 修回日期:2006-10-13

作者简介:

貌学良(1965-),男,硕士,高级工程师,从事软件开发及计算机教学工作。E-mail:mao_xl@163.com