

# 基于面向对象的智能继电保护装置的软件设计

蔡树立

(桂林电子工业学院,广西 桂林 541004)

摘要:提出了以数据为基础的面向对象技术应用在智能继电保护装置中的新思路,详细介绍了这种思路的工作原理、实现过程、应用实例和功能特点。

关键词:数据;面向对象;继电保护;智能

中图分类号:TM774 文献标识码:A 文章编号:1003-4897(2005)06-0059-03

## 0 引言

传统继电保护装置的设计方法是:在设计人员对保护方案深入理解后,将其以逻辑的形式在程序中以逻辑的方法形成<sup>[1]</sup>,采用的是结构化的系统设计方式。其设计思路是以保护功能为线索,即现场需要什么保护,就在程序中实现什么功能;相对先进的思路是将所有可以考虑的保护功能全部实现,到现场后再根据实际情况组态。这种设计思路便于专业人员的理解和维护,却降低了系统的开放性、可扩展性和可移植性。

面向对象的设计理念,其“对象”是现实世界中具体事物和概念的抽象,开发人员和用户彼此容易沟通;在系统的设计和实现阶段,通过封装、继承、多态等手段,可以大大降低系统实现的复杂度,提高代码的可重用性、可靠性及可维护性,从而加速了问题求解过程。正是由于面向对象技术的各种优点,该方法已经在操作系统、数据库管理、作战模拟、计算机网络与通信、人机界面设计等方面获得了广泛的应用<sup>[2]</sup>。

本文提出了基于面向对象的设计思路,将电力系统继电保护装置的数据抽象化,将不同保护功能的操作原理和操作过程抽象成数据对象的属性和方法,充分利用面向对象技术封装和继承的优点,提高了保护装置的可靠性、安全性和可扩展性。

## 1 面向对象的思想在智能继电保护设计中的应用

### 1.1 对继电保护对象的抽象思路

在继电保护装置中,可以有两种不同的抽象思路。一是基于继电保护功能的抽象化,例如:建立继电保护基类 T\_Base\_Relay,基类的属性包括继电器名字、继电器类型、继电器状态等;将继电保护的通

用处理模块,比如:选相模块、方向判断模块、计算相间阻抗模块、启动模块、出口信号模块等设置为基类的方法。这样,最高级的继电保护对象即抽象为 T\_Base\_Relay,相当于面向对象编程环境 Delphi 中的 TObject。对于具体的继电保护功能,比如:高频保护、距离保护、零序保护、高频方向保护等都可以从 T\_Base\_Relay 继承<sup>[3]</sup>。

在继电保护装置中,还有一种抽象的思路:以数据为对象。以数据为对象的抽象思路有利于整个装置的模块化和智能化,本文即采用了这种新的思路。

继电保护种类繁多,原理各异,但一般继电保护装置都由测量单元、逻辑单元和执行单元组成。测量单元测得保护对象的电流、电压值,经 CT、PT 变换进入逻辑处理单元;逻辑单元根据用户设定的逻辑时序关系,与定值进行比较,以判断是否动作、如何动作;执行单元具体执行,比如保护出口报警或者操作断路器等。由继电保护的这 3 个基本过程可以看出,如果把继电保护装置看作一个黑匣子,其实质是一个数据传送的过程,在整个系统中,设计人员和用户都可以不管系统到底要实现什么功能,将其具体功能的实现看作是数据在系统中逻辑处理的出口。将系统数据抽象为对象,对象的不同组合看作是对对象的属性和方法。这样不仅可以方便地组合出各种已知的继电保护功能,而且对那些在设计阶段无法确定,需要根据用户实际现场灵活配置的继电保护功能以及其它监控功能,通过系统数据的不同组合进行现场组态。

### 1.2 类的定义

一般的保护监控装置,整个系统所用到的数据可以分为 3 种类型:遥测量、遥信量以及出口信号量。其中遥测量主要指电气信息量,包括电压、电流等;遥信量主要指记录断路器、隔离开关、继电器以及各种传感器的开关信号(0 或非 0)等;出口信号

主要指出口继电器或者表征显示屏、调试口、上级配电自动化主站等通讯地址<sup>[4]</sup>。众所周知,上述3种数据都仅仅是最底层的数据元素,真正在电力系统继电保护中操作的对象往往是线路、变压器、FACTS设备等。因此在定义抽象类时,可以确定4种数据对象类,分别为:T\_YC\_ELEMENT、T\_YX\_ELEMENT、T\_YK\_ELEMENT、T\_PROTECT\_OBJECT。

总结这4种数据对象类,还可以抽象出它们共同的属性和方法,建立真正的继电保护装置的数据基类T\_BASE\_DATA。基类T\_BASE\_DATA封装属性包括:序号、名称;封装方法包括:数据保存、数据通讯、数据出口等。其中“序号”是系统内数据的统一编码,具有系统唯一性;“名称”是对数据的描述,便于人机对话的交流;“数据保存”是指进行历史数据保存,至于保存的算法和地点都可以通过方法中的参数传递;“数据通讯”的通讯协议和通讯地址可以通过参数传递,而通讯协议的具体内容需要定制在基类中;“数据出口”包括保护出口报警或者操作断路器等。

T\_YC\_ELEMENT、T\_YX\_ELEMENT、T\_YK\_ELEMENT、T\_PROTECT\_OBJECT是由基类T\_BASE\_DATA继承而来的,但同时又有功能的扩展,这些子类分别定义如下:

遥测量T\_YC\_ELEMENT,其属性除了继承父类属性以外,还扩充了“计算数值”属性;其方法除了继承父类方法以外,还扩充了“快速傅里叶计算”、“均方根求值”等。其中“快速傅里叶计算”可以对遥测量进行FFT计算,其计算结果包括各次谐波的幅值、有效值、实虚部等;“均方根求值”是为了电能质量参数计算的需要。遥测类的这两个方法所计算的结果都保存到“计算数值”属性中,“计算数值”是一个数组。

遥信量T\_YX\_ELEMENT,其方法除了继承父类方法以外,还扩充了“遥信预处理”、“变位SOE记录”等。

保护基对象T\_PROTECT\_OBJECT,其方法除了继承父类方法以外,还扩充了“故障选相”、“功率计算”、“阻抗计算”、“越限检测”、“逻辑处理”等方法。

由上述对象的定义可以看出,遥信、遥测和出口信号都是基本的数据元。一方面,遥测、遥信量在自己预处理后,可以直接出口,比如某个电流量的幅值按一定算法自动保存、某个开关量变位报警等;另一方面,它们也可以作为保护基对象方法处理中的传

递参数,供保护基对象以及它的子类使用。因为出口信号对应的现实对象是出口继电器、通讯地址等,因此它也可以在系统通讯、出口时被遥测、遥信或者保护基对象类作为对象调用。

这种对象类的定义可以用一个形象的比喻来形容:“遥信”、“遥测”和“出口信号”都相当于人体的基本神经元,它们自身可以完成一定的人体功能;同时由多个神经元组织在一起,又形成了人体的某个器官(保护基对象及其子类),可以完成一些特定的功能。

### 1.3 类定义的软件实现

以保护基对象类T\_PROTECT\_OBJECT为例,如果利用面向对象的编程环境DELPH实现其类定义,核心程序如下:

```
Type
    T_PROTECT_OBJECT = Class (T_BASE_DATA)
Public
    Property No;                                (继承基对象属性)
    Property Name;
    Function DataSave ();                       (继承基对象方法)
    Function DataCommu ();
    Function DataOutput ();
    Function short SelectPhase (Complex Ia, Complex Ib,
    Complex Ic);                                (故障选相)
    Function void PowerCal (Complex I, Complex U);
                                                (功率计算)
    Function void ZCal (short FaultPhase, Complex Temp0,
    Complex Temp1, Complex Temp2, Complex Temp3);
                                                (阻抗计算)
    Function short InRange (Complex V set, Complex Vcal);
                                                (越限检测)
    Function Void LogicalDeal ();              (逻辑处理)
End;
```

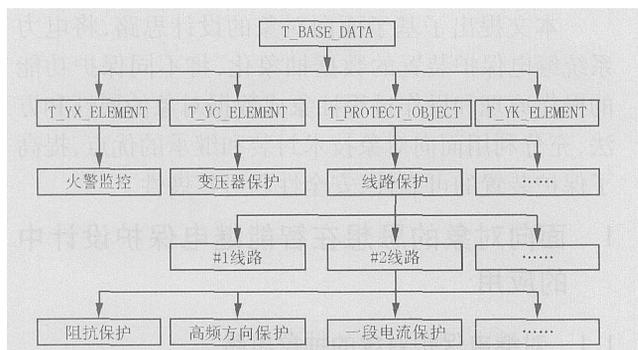


图1 继电保护装置中对象继承关系树

Fig 1 Inherited relation tree of object in the relaying protection device

如果继电保护装置是通过单片机构建的,以上的类定义也可以通过 C、汇编编程实现。当然,其具体的实现方法跟高级语言有些不同,主要是通过函数和宏的调用来实现代码的重用。

#### 1.4 继电保护装置的功能实现

图 1 是继电保护装置对象继承关系树。如图 1,如果要想实现 #2 线路的阻抗保护,可以把该保护功能看作一个实例 E\_Line2\_Zprotect,其属性和方法都是由父对象继承而来。子类从基类中继承各种属性和方法,从而提高了系统的可重用性和可维护性。为了适应现场实际的不同需要,不同子对象在接受同一父对象的同一个消息时,会作出不同的响应(即一个消息,多种行为),这正是面向对象设计思想中“多态性”的体现。多态性的使用不仅有助于编程人员进行抽象化思维,而且还为该软件的扩充提供了极大的灵活性。同时,在子对象的实例过程中,允许修改或者添加新的属性和方法,使得面向对象的继电保护装置具有很强的智能化特点,能够自由实现功能的扩展,具有更广泛的现场适应性。

#### 1.5 说明

本文提出的面向数据进行抽象的思路,在具体实现过程中,还可以有不同的实现方法。比如,对于基类 T\_BASE\_DATA,可以把数据保存、数据通讯和数据出口这 3 个方法合并为一个方法:数据处理;在图 1 的对象继承关系树中,对于 T\_PROTECT\_OBJECT 的子类层次关系也可以有不同于本文的划分方式。总之,以数据为对象的抽象思路,其实现过程并不是唯一的,还可以进一步探讨优化。

## 2 结论

利用上述思路,本文完成了“新一代智能型环网柜保护监测装置”的研究设计。开发实践证明,由这种思路设计的软件,其灵活性、开放性、可维护性和可扩充性都比传统程序设计方法有了较大提

高。实际上,基于数据的面向对象的设计思想,其应用不仅仅限制于继电保护装置,如果抽象合理,构思科学,还可以在潮流计算、变电站综合自动化等电力系统多个领域得到广泛应用。

#### 参考文献:

- [1] 王威,郁惟镛,张沛超.面向对象的继电保护整定计算专家系统的研究[J].电力系统及其自动化学报,2001,13(2):1-5.  
WANG Wei, YU Wei-yong, ZHANG Pei-chao. Research on Object-oriented Expert System of Relay Setting Calculation[J]. Proceedings of the EPSA, 2001, 13(2): 1-5.
- [2] 陆进军,黄家裕.基于面向对象编程的电力系统潮流计算软件[J].电力系统及其自动化学报,2000,12(1):32-35.  
LU Jin-jun, HUANG Jia-yu. Power Flow Calculation Software of Electric Power System Based on Object Oriented Programming[J]. Proceedings of the EPSA, 2000, 12(1): 32-35.
- [3] 杨永生.基于面向对象技术的微机保护仿真[J].电力自动化设备,2001,21(7):46-48.  
YANG Yong-sheng. Simulation of Microcomputer Based on Object Oriented Technology[J]. Electric Power Automation Equipment, 2001, 21(7): 46-48.
- [4] 董海鹰,白建社,薛钧义.变电站远程监控系统体系结构及面向对象数据库研究[J].电力系统及其自动化学报,2002,14(6):13-17.  
Dong Hai-ying, BA Jian-she, XUE Jun-yi. Study on System Structure and Oriented-objected Data Base of Substation[J]. Proceedings of the EPSA, 2002, 14(6): 13-17.

收稿日期: 2004-06-29; 修回日期: 2004-10-31

#### 作者简介:

蔡树立(1975-),男,讲师,研究方向为电力系统配电网自动化、GIS/MIS在电力系统中的应用。E-mail: caishuli@263.net

## Software design of intelligent relaying protection device based on object-oriented technology

CAI Shu-li

(Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** A new idea about data-based object-oriented technology applied in intelligent relaying protection device is put forward. And the operation principle, process, applied example and function characteristic of the proposed idea are detailed.

**Key words:** data; object-oriented; relaying protection; intelligent