

面向对象的电力系统图形程序设计方法

刘伟, 许珉, 杨宛辉

(郑州大学工学院电气工程学院, 河南 郑州 450002)

摘要: 从实用角度出发, 介绍了面向对象的电力系统图形程序的设计技术和方法, 具有参考和使用价值。

关键词: 面向对象程序设计; 多态性; 电力系统; 图形界面

中图分类号: TM769 文献标识码: A 文章编号: 1003-4897(2003)11-0044-04

1 引言

面向对象程序设计(Object - Oriented Programming, 以下简称 OPP)并不是一种程序设计语言, 而是一种新的概念, 即关于数据、过程和它们之间关系的一种新的思维方法, 它是一种试图摹仿我们建立现实世界模型的程序设计方法, 为适应现实生活的复杂性, 我们已经逐渐具备了很好的概念来形成分类和抽象能力。在我们的词汇中几乎每个名词都表示一类事物, 享有一组属性或行为特征。OPP的发展正是利用了我们事物分类和抽象这样一种自然的倾向。类是 OOP 技术最基本的概念之一, 而多态性和封装性是类的基本属性, 多态性是指给行为起一个名字或符号, 它共享一个类的层次, 在这个层次中的每个类都以适合自己的方式实现这个行为, 也就意味着以同样的方法可以处理多种不同类型的函数。图形化的电力系统计算软件, 不需要用户对接线图进行节点编号, 界面友好, 输入方便, 使用简单、直观, 不易出错, 深受使用者欢迎。如本文编写的一个电力图形系统, 图形系统中包含着电力系统各种电气元件及文本框等各种图形元素, 对图形元素能够实现绘制、选中、撤消、存贮等各种操作, 如果考虑一个图形系统所有的功能, 不久就会把思路搞乱, 但是如果把图形系统看作是一组独立的对象的集合, 问题就容易解决得多。例如可以把一个电力系统图形分成以下主要对象: 发电机、变压器、断路器等, 它通过将数据和处理数据的函数封装到一起, 并通过函数与外部进行联系, 实现系统设计的高度结构化。例如将发电机作为一个对象, 在这个对象中不仅包含了发电机数据结构, 而且包含了对发电机的各种操作(如绘制、选择等), 这样就把发电机这个

对象的数据资料和处理功能融合在了一起。面向对象的电力系统图形程序设计方法充分利用了对象使用时提供的以下优点:

- (1) 设计和代码易于重用, 程序可扩性强;
- (2) 封装性增强, 提高了数据和程序的可靠性;
- (3) 减少了程序设计的代码量, 简化了程序结构。

本文采用 VB6.0 开发了面向对象的电力系统图形软件, 并在电力系统短路计算与电气设备选择校验计算程序中进行了应用。

2 面向对象的电力系统图形程序设计

2.1 主要设计思路

电力系统图形元件的绘制, 可以采用位图, 由于各元件的位图在绘图区拼装而组成图形元件的变换较困难, 本文采用绘图函数(在图元编辑中引入计算机图像处理的有关方法), 建立元件类, 可以方便地实现图形元件变换。图形元件的存贮一般可以采用文件和数据库两种方式, 编辑图形时, 由于编辑绘制速度较慢, 本文采取直接将图元信息存入数据库的方法, 充分利用 VB 对数据库的查找、增加、删除、修改语句, 操作十分方便。图元编辑包括图元绘制、旋转、移动、删除、修改颜色、放大、缩小、文本输入等。图元设备参数输入及修改包括断路器、输电线路、双绕组变压器、三绕组变压器、自耦变压器、电抗器、发电机等, 参数还包括断路器的合、断信息, 以便使用者改变运行方式; 中性点是否接地(对于变压器、发电机), 以便形成零序网络。绘图完成以后, 自动形成节点编号, 自动进行接线拓扑分析, 去掉零阻抗支路和断路器支路, 完成参数计算。绘图是在 Picture1 控件上进行的, 图形可以滚动, 绘图界面见图 1。

2.2 面向对象的程序设计方法

2.2.1 图元的设计

VB6.0 没有继承机制, 在进行面向对象的程序

基金项目: 河南省自然科学基金项目(004061300)

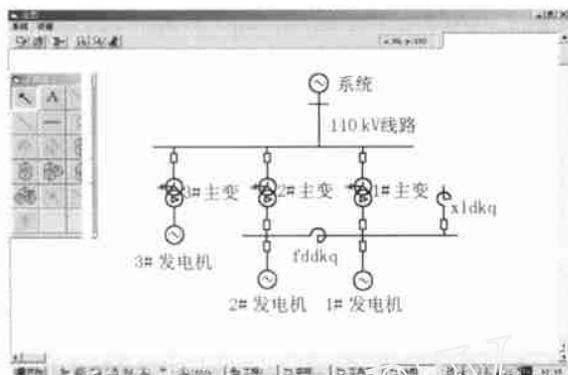


图 1 绘图界面

Fig. 1 Graph interface

设计时,可以采用多态性,用抽象类和具体类,还可以起到其它语言中继承的作用。进行图元的绘制时,面向对象的程序的设计方法,首先设计定义图元类,图元类的抽象类是绘制图元时调用的通用类,不含任何实现代码,其用途是为添加到其他类中的接口提供模板,为了在具体类中实现抽象类的接口,需要用到 Implements 语句,抽象类的具体定义如下:

类名:Drawobject

```
Public Sub draw(ByVal X1 As Integer, ByVal y1 As Integer,
ByVal X2 As Integer, ByVal Y2 As Integer, ByVal c1 As Integer,
ByVal c2 As Integer, ByVal c3 As Integer, ByVal z1 As Integer,
ByVal z2 As Integer)
```

```
End Sub
```

```
Property Get X1() As Integer
```

```
End Property
```

(读属性过程)

```
Property Let X1(Cnew As Integer)
```

```
End Property
```

(写属性过程)

```
Property Get Y1() As Integer
```

```
End Property
```

```
Property Let Y1(Cnew As Integer)
```

```
End Property
```

(其它变量的属性过程类似)

具体类的定义如下:

类名: Generator (发电机)

定义私有变量(属性):

```
Private mX1, mY1 As Integer
```

```
Private mXr, mYr As Integer
```

```
Private mXe, mYe As Integer
```

```
Private mX1, mY1, mX2, mY2, mX3, mY3 As Integer
```

Implements DrawObject (为了在具体类中实现抽象类的接口,需要用到 Implements 语句)

定义公有函数(方法):

```
Public Sub DrawObject-draw(ByVal X1 As Integer, ByVal y1
```

```
As Integer, ByVal X2 As Integer, ByVal Y2 As Integer, ByVal c1
As Integer, ByVal c2 As Integer, ByVal c3 As Integer, ByVal z1
As Integer, ByVal z2 As Integer, ByVal frm As Form)
```

.....
(具体程序代码可参考下文的例子:发电机元件绘制程序)

属性过程定义如下:

```
property Get DrawObject- X1() As Integer
```

```
DrawObject- X1 = mX1
```

```
End Property
```

```
Property Let DrawObject- X1(Xnew As Integer)
```

```
mX1 = Xnew
```

```
End Property
```

```
Property Get DrawObject- Y1() As Integer
```

```
DrawObject- Y1 = mY1
```

```
End Property
```

```
Property Let DrawObject- Y1(Cnew As Integer)
```

```
mY1 = Cnew
```

```
End Property
```

.....

断路器图元类:

类名:QF

```
Private mX1, mY1 As Integer
```

```
Private mXr, mYr As Integer
```

```
Private mXz, mYz As Integer
```

```
Private mXs, mYs, mXe, mYe, mXm, mYm As Integer
```

```
Implements DrawObject
```

```
Public Sub DrawObject-draw(ByVal X1 As Integer, ByVal y1
As Integer, ByVal X2 As Integer, ByVal Y2 As Integer, ByVal c1
As Integer, ByVal c2 As Integer, ByVal c3 As Integer, ByVal z1
As Integer, ByVal z2 As Integer)
```

.....
(断路器图元类属性过程的定义与发电机图元类的属性过程的定义类同)

多态性的实现:

定义类对象

```
Private g1 As Generator
```

```
Private qf1 As QF
```

定义类对象数组

```
Private draw1(30) as DrawObject
```

```
Set g1 = New Generator
```

```
Set qf1 = New QF
```

```
Set draw1(1) = g1
```

```
Set draw1(2) = qf1
```

在所有需要画图元的事件中,调用 Call draw1 (1x). draw(x1, y1, x2, y2, c1, c2, c3, z1, z2) 即可。其中 1x 为图元代码,当 1x = 1 时,画发电机;当 1x = 2 时,画 QF... 等等,画所有图形元件用的是同一条

语句。如不使用多态性,在所有需要绘制图元(如鼠标的 MouseDown、MouseUp、MouseMove 等)事件中以及编辑各功能程序中,一般要用 Select Case 语句进行分支处理,程序复杂,而采用多态性,就无须编写烦琐的程序代码。由以上叙述可知其最大的优点是程序可扩充性好,增加新的图元,只需增加新的类,在 Formload 中定义新的类对象,然后简单修改程序即可。图元信息的存贮也很方便,读出坐标 X1, Y1, 用 Draw1(1x).x1 和 Draw(1x).y1。

2.2.2 窗体的多态性

由于各种电气设备其参数不同,参数输入也需采用不同的窗体,采用多态性可使参数输入程序简单,其实现方法如下:

定义窗体变量数组:

```
Dim myforms(30) As Form
Set myforms(1) As Generatorcs
Set myforms(2) As Qfcs
```

如果 lx = 1,调用 myforms(lx).show 1 即可显示输入发电机参数的窗体。与画图形元件一样,显示所有参数输入的窗体使用的是同一条语句。它与类的多态性配合,使程序大为简化。

2.2.3 数据库模块的功能及实现

图元信息直接存入数据库,本文利用 Access 数据库文件作为数据源,使用 SQL 语言作为操作、查询语言,依照 ADO 数据库访问标准,实现了完整的数据库操作功能。其中图元信息存放的表结构如下:

图元信息表

字段	数据类型
type	integer ;元件类型标志
x1,y1	integer ;元件起点坐标,也是旋转参考点。
x2,y2	integer ;两端点元件终点坐标
x3,y3	integer ;三端点元件第三点坐标
c1	integer ;颜色 1。
c2	integer ;颜色 2,用于两电压等级元件
c3	integer ;颜色 3,用于三电压等级元件
xe,ye	integer ;旋转第二参考点坐标
xl,yl	integer ;元件选中区左上角坐标
xr,yr	integer ;元件选中区右下角坐标
zon1	integer ;开关、发电机变压器中性点接地状态
zon2	integer ;变压器中性点接地状态

图元信息存放在数据库中,结构清晰,查询方便,调用时打开数据库,将图元记录读出,根据图元类型及坐标即可将图形绘出显示出来。

2.2.4 图元中元件的绘制

在图形元件的绘制过程中,关键是图形变换。主要包括:平移,旋转,比例及复合变换等。最复杂的,也是最主要的是旋转变换。下面主要介绍一下旋转变换的实现。

(1) 旋转变换

旋转是以某个参考点为圆心,将对象的各点(x, y)围绕圆心转动一个角度(顺时针旋转角为负值,逆时针旋转角为正值),变为新的坐标(x', y')

当参考点为(0,0)时,旋转的公式为:

$$\begin{aligned}x' &= r * \cos(\theta + \alpha) = r * \cos \theta * \cos \alpha - \\ & \quad r * \sin \theta * \sin \alpha \\ y' &= r * \sin(\theta + \alpha) = r * \sin \theta * \cos \alpha + \\ & \quad r * \cos \theta * \sin \alpha\end{aligned}$$

因为 $x = r * \cos \theta$, $y = r * \sin \theta$,所以上式可以化为:

$$\begin{aligned}x' &= x * \cos \alpha - y * \sin \alpha \\ y' &= y * \cos \alpha + x * \sin \alpha\end{aligned}$$

如果参考点不是(0,0),而是任意一点(xt, yt),那么,绕(xt, yt)点的旋转由三个步骤完成:

- 将对象平移 $Tx = -xt$, $Ty = -yt$
- 按上式作旋转变换
- 平移 $Tx = xt$, $Ty = yt$

组合这三个步骤的计算公式为:

$$\begin{aligned}x' &= xt + (x - xt) \cos \alpha - (y - yt) \sin \alpha \\ y' &= yt + (y - yt) \cos \alpha - (x - xt) \sin \alpha\end{aligned}$$

(2) 元件任意角度旋转的实现方法

元件任意角度旋转是用元件的(x1, y1)坐标为起点,也是在画元件时鼠标的第一个点,以鼠标的起点到鼠标最终位置坐标(x2, y2)连接的直线为元件图形的 Y 轴,旋转角度的计算如下:

$$\begin{aligned}\cos \alpha &= \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ \sin \alpha &= \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}\end{aligned}$$

画元件时,拖动鼠标旋转到需要位置(x2, y2)即可,或用元件 Y 轴上的某一点,作为绘图旋转的第二参考点存入图元表。结合所做的工作,下面举一发电机元件绘制程序为例来说明:

```
Private mX1, mY1 As Integer
Private mXr, mYr As Integer
Private mXe, mYe As Integer
Private mX1, mY1, mX2, mY2, mX3, mY3 As Integer
Implements DrawObject
Public Sub DrawObject-draw(ByVal X1 As Integer, ByVal y1
```

```

As Integer, ByVal X2 As Integer, ByVal Y2 As Integer, ByVal c1
As Integer, ByVal c2 As Integer, ByVal c3 As Integer, ByVal z1
As Integer, ByVal z2 As Integer
    Dim L As Single
    Dim kc As Single
    Dim Ks As Single
    If X2 - X1 <> 0 Or Y2 - y1 <> 0 Then
        '计算旋转角度'
        L = Sqr((X2 - X1) ^ 2 + (Y2 - y1) ^ 2)
        kc = (X2 - X1) / L
        Ks = (Y2 - y1) / L
    Else
        kc = 0
        Ks = 1
    End If
    Const PI = 3.14159
    '画发电机引出线直线,颜色由 c1 决定'
    frm.Picture1.Line (X1, y1) - (X1 + 20 * kc, y1 +
    20 * Ks), QBColor(c1)
    '画半径为 13 个像素的圆'
    frm.Picture1.Circle (X1 + 33 * kc,
    y1 + 33 * Ks), 13, QBColor(c1)
    '画“~”符号,由两个半圆组成'
    frm.Picture1.Circle (X1 + 33 * kc - 4,
    y1 + 33 * Ks), 4, QBColor(c1), 0, PI
    frm.Picture1.Circle (X1 + 33 * kc + 4,
    y1 + 33 * Ks), 4, QBColor(c1), PI, 2 * PI
    mXl = X1
    mYl = y1
    mX2 = 0
    mY2 = 0
    mXe = X1 + 46 * kc
    mYe = y1 + 46 * Ks
    mXl = X1 + 33 * kc - 13
    mYl = y1 + 33 * Ks - 13
    mXr = X1 + 33 * kc + 13

```

```
mYr = y1 + 33 * Ks + 13
```

```
End Sub
```

mXl, mYl : 发电机元件起点坐标和旋转参考点坐标, mXe, mYe : 旋转第二参考点坐标, mXl, mYl : 选中区左上角坐标, mXr, mYr : 元件选中区右下角坐标。各坐标均为变换后的值, 这些参数由读属性过程读出存入图元表。

3 结束语

本文对面向对象的电力系统图形程序进行了研究, 从实用角度出发, 详细介绍了图元类及其多态性的设计技术和方法, 具有很大的实用性。按本文设计的电力系统图形程序, 已经在电力系统短路电流计算和电气设备选择校验计算程序中使用, 具有很好的实用价值。

参考文献:

- [1] 廖彬山, 黄维通, 等. Visual Basic 中文版面向对象与可视化程序设计[M]. 北京: 清华大学出版社, 2001.
- [2] 闫自凯, 杨宛辉, 等. 电力系统智能图形支撑平台的研究[J]. 继电器, 2001, 29(4): 26 - 28.
- [3] 潘云鹤. 计算机图形学——原理、方法及应用[M]. 北京: 高等教育出版社, 2001.
- [4] 陈建春. Visual C++ 高级编程技术——开发实例剖析[M]. 北京: 电子工业出版社, 1999.

收稿日期: 2003-01-27

作者简介:

刘伟(1976-), 女, 硕士研究生, 研究方向为电力图形程序设计;

许珉(1956-), 男, 副教授, 从事电力系统监视与控制方面研究;

杨宛辉(1943-), 女, 教授, 从事电力系统监视与控制方面研究。

Method of object - oriented graphic programming in power system

LIU Wei, XU Min, YANG Wan-hui

(School of Electric Engineering, Zhengzhou University, Zhengzhou 450002, China)

Abstract: From the viewpoint of application, a new kind of object - oriented programming in power system is introduced. It has much practical value.

Key words: object - oriented programming; polymorphism; power system; graph interface

作者补遗:

本人在《继电器》2002年第11期上发表的《基于零序功率的小电流选线方法》, 文中有关零序有功功率的判据, 参考了西门子公司7SI系统产品的相关资料, 原文的参考文献中未能反映, 特此补充。

作者: 杨汉生