

利用 Delphi 自定义控件类实现通用数据控制程序

——基于数据字典的界面动态生成技术

曲阜平 (后勤工程学院指挥自动化教研室 400042)

王晓武 (中国人民解放军总后勤部司令部 100842)

摘要:本文论述了面向对象技术在 MIS 系统开发中的应用及利用 Delphi 提供的功能建立自定义控件类,在程序运行中动态生成与释放控件对象,控件实例数组的使用以及如何在 Delphi 下编写面向数据字典的通用数据编辑和浏览界面。

关键词:面向对象(OOP) 数据字典 动态生成

1. 前言

基于 Windows 的可视化开发工具的种类越来越多,使得开发人员经常感到无所适从,本文不想就孰优孰劣妄下断言,仅将我们在用 Delphi 作为前台开发工具,用 ORACLE 作为 DBMS 进行 MIS 系统开发过程中所得到的一点经验体会拿来与各位应用程序开发人员共享,希望能对大家有所帮助。

Delphi 提供的 TDBGrid、TQuery、TTable、TColumns、TField 等与数据对象相关的控件类,使开发基于 Client/Server 体系结构的 MIS 系统更加方便、快捷;它对拖——放编程的支持,使得对数据表、视图等的处理更加直观。但由于不同的表、视图等对象结构各不相同,通常针对某一表的数据处理(增、删、改、查)不得不编写一段特定的应用程序。虽然这种可视化编程比以往的编程方法方便,但仍然存在两个方面的问题:一方面在大型复杂的 MIS 系统中,表的数量一般都比较复杂,即使是一名熟练的程序员,也要花较长的时间才能完成对应特定表的编程;二是为某一应用所开发的系统(例如为人事部门开发的人事管理系统)在别的应用中(例如生产管理)几乎没有任何使用价值。

也就是说,采用这种通常的编程方法,Delphi 不能实现脱离特定表结构编程,代码也无法重用,这样使得 MIS 系统的编程效率降低。仔细分析,这两个问题是能够解决的,虽然不同的表结构不同,但对它们所进行的处理却是完全一样的:不外乎增、删、改、查,同时它们也有许多共同之处:都是由字段对象组成,具有相同的约束集和字段类型集。能否通过对表结构进行统一描述,使得对表的处理能够独立于表结构?为此在某 MIS 系统开发中,

我们引入了数据字典来描述表的结构及属性(字段名、类型、长度等),开发出了面向字典的通用应用程序,不仅可以用其处理同一系统中的不同表,而且可以方便地将其用于另一应用系统中,要做的只是在表结构字典中对要进行处理的表的结构和属性进行必要的描述。

Delphi 控件是可视构件的总称。所有的控件都具有可视方面的共同属性、方法和事件,如位置、光标、绘制以及移动的方法、反应鼠标动作的事件等。控件的共同属性、方法和事件都被封装在控件的抽象基类 TControl 类中。从面向对象的观点来看,每一类控件都是 TControl 类的一个子类,它们在继承了 TControl 类全部成员的同时定义了自己特有的成员,并声明了部分 TControl 类的保留成员,从而使其具有自己的特性。Delphi 为可视化编程提供了丰富的控件,满足了软件开发的基本需要,但对于一些复杂软件的特殊开发需求,Delphi 提供的基本控件则难以满足要求。Delphi 的所有控件都是用一种完全面向对象的语言——Object Pascal 编写的,并提供了源代码,因此,完全可以根据使用的需要以某个控件类为基础来定义自己的控件类,使用继承、重载和增加新成员等机制,建立起适合特定需要的控件类,为编程提供方便。数据字典结合控件数组及自定义控件类,不仅可简化编程,解决上述问题,而且使应用程序具有了更大的灵活性和用户友好性。

2. 编写自定义控件类

充分利用 Delphi 面向对象的强大功能,对控件类库进行适当扩充,能够起到事半功倍的效果。在开发 MIS 系统时,经常要用到形如图 1 所示的部件组合(其中的“数据编辑(浏览)框”直接与数据库中的某数据表的一个

字段相接),其简单示例如图2。通常是使用 Delphi 的某几类基本控件的对象组合成一个如图1的部件,最后产生一个类似图2的示例,这种方法的主要不足在于:当界面实时生成时,部件的引用与动态随机定位编程困难。实时产生如图2所示的显示界面需要对涉及的所有控件进行寻址与重定义,最简单的情况是图1的部件由两个控件对象组合而成,那么改变该部件的位置、尺寸、标题、显示内容等,就要对两个控件单独进行寻址和定义。Delphi 并不直接提供控件数组功能,寻址不易实现;而且由于对同类控件的处理重复性编程量较大,使出错的可能性也大大增加。

笔者针对上述问题,自定义了一个控件类:TDBEditPanel,在实践中使用效果很好。其样式如图1,利用 Delphi 的 component 菜单项中的 Install component 子项将该控件类加到控件模板的 Samples 页上,就可象使用 Delphi 自带的控件一样使用它了。

标题	数据编辑(浏览)框
----	-----------

图 1

单位	级别	编制员额	现有员额	现有车辆
后勤处	团	2121212	1111111	12090

图 2

3. 数据操作界面动态生成技术

高质量 MIS 系统必须向用户提供最大的灵活性,其中最基本的一点是:在对数据进行浏览、编辑时,既要有列表方式,又要有卡片方式,不仅允许用户设置查询条件选取记录,而且允许选择部分字段(域)查看、编辑,因为用户的选择是不确定的,所以格式固定的界面已不能满足要求。列表和卡片在程序运行过程中实时产生和处理涉及到两个问题:一是依据问题,即控件以什么样的模式出现?其根据是什么?二是对控件的寻址和重定义问题。

所谓的依据是指系统以什么样的方式记录用户的选择,并以此为根据来产生所需的列表及卡片。在我们设计的系统中采用面向数据字典的方法,极大地提高了程序模块的可重用性,使用的字典(DD)基本结构如下:

项名	类型	中文名	含义
DD01	C(8)	数据表名	
DD02	C(8)	数据项名	字段名
DD03	C(8)	数据项中文名	字段中文名
DD05	C(8)	类型	
DD06	N(3)	长度	
DD09	C(1)	选用标记	Y:选中,N未选中
DD11	C(1)	代码类型	分层、不分层
DD12	C(8)	代码表名	
DD13	C(8)	代码项名	
DD14	C(8)	代码项中文名	

数据字典使用的基本思路是,根据用户的选择在字典 BZ-DD 的参于操作标记字段 DD90 上做标记,然后以字典 DD 为依据来生成控件并确定其相关属性。Delphi 的数据表格控件 DBGrid 为列表方式提供了极大的方便,这方面内容的本文不赘述。卡片的处理涉及到控件的寻址、重定义和事件处理等问题,采用控件数组可使编程大大简化。控件数组在 Visual Basic 等开发环境中的应用极其普遍,而在 Delphi 中却属于应用较少的功能,与此相关的资料很罕见。我们编制了一个展示面向数据字典的卡片动态生成、控件寻址以及控件数组的使用的例程,该例程在 windows95 下调试通过。

(编者注:由于程序清单较长,本刊略,有兴趣的读者可与作者联系。)

应用中首先需要注意的问题是:不要把控件的属性 Owner 和 Parent 搞混淆了。Owner 属性决定控件的存储和存在,从 TComponent 类继承而来,在一个控件的实例产生时就要指定,如果控件的 Owner 被释放掉(所占内存被释放),该控件也就不存在了。Parent 属性决定的是控件的显示窗口,继承于 TControl 类,该属性在程序中多处使用同一个控件时特别有用。例如想在多页控制中使用同一个状态条,就可根据当前哪页处于活动状态,将其指定为所用状态条的 Parent(如:StatusBar1.parent:=PageControll.ActivePage),就可实现共用一个状态条的目的。

在使用控件数组时可能要涉及到在程序运行过程中为某一控件指定事件处理过程,方法是把相应的事件处理句柄指定为某一已定义的过程,例如要让过程 Click1(Sender:Tobject)处理按钮 Button1 的单击事件,就须使 Button1.Onclick:=Click1;如果数据编辑框 DBEdit1 的双

击事件也要由同一个过程进行处理,则须把 DBEdit1 的事件处理句柄指定为 Click1,即 DBEdit1.OnDbClick:=Click1。

4. 结束语

实时动态生成控件增强了程序的灵活性;控件数组的使用避免了编写大量重复代码,减少了程序出错率。

数据字典的使用使程序可重用性大大增强,提高了代码的重用性,使开发通用应用程序变得更加方便快捷。

Delphi 是一种完全面向对象的可视化开发工具,具有很强的灵活性,对于开发需求变化较多、界面复杂多变的 MIS 系统有很大的优势。如能灵活应用其面向对象机制,可以大大简化编程,并使应用系统具有更大的用户友好性。

MIS 系统开发主要有两种不同的方法:面向数据的方法和面向过程的方法。用第二种方法开发出的应用系统,一般在使用的初始阶段能够较好地满足应用的需要,

获得较好的性能,随着应用的发展变化,要进行较大的改动和维护。用第一种方法开发出的系统能够比较合理、自然地模拟一个单位。一般说来,一个单位的数据总是相对稳定的,而处理则是变动的,因此用第一种方法开发的系统也就相对稳定,本文介绍的方法虽然适用于以上两种方法,但应用于第二种将更有效。

参考文献

- [1] Marco Cantu 著,王辉 谭海平等译《Delphi3 从入门到精通》电子工业出版社
- [2] 刘文智 郭文健著《Delphi 2.0 入门与提高》清华大学出版社
- [3] 周军民 姚庭宝等编著《精通 Delphi》电子工业出版社

(来稿时间:1998 年 6 月)