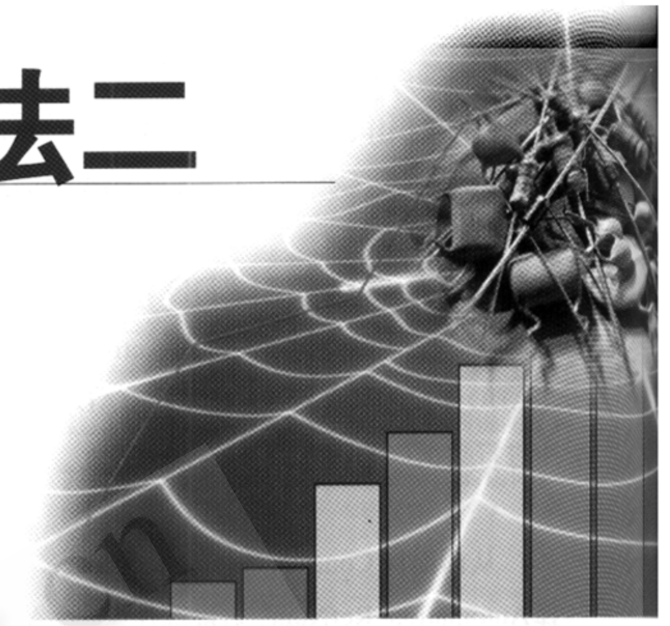
创建动态图表方法二



王有一（西安国家电力公司热工研究院 710032）

摘要：本文介绍使用内存数组作缓冲工作区创建动态图表的方法，它较之基于数据库的动态图表，运行效率更高。同时，通过应用实例介绍怎样更合理地优化程序设计。

关键词：内存数组　缓冲工作区　数组指针

1　前言

在《创建实时数据库动态图表》一文中，笔者介绍的创建动态图表的方法需要设置一个临时数据库作缓冲区，这将增加程序访问磁盘的次数，从而降低系统运行效率。本文介绍的另一种方法是用内存数组代替磁盘数据库作图表的缓冲工作区，在Windows98/NT多任务运行环境中，它的运行效率更高。

2　非数据库图表

创建数据库图表使用TDBChart构件，创建非数据库图表应选择TChart构件，因为它无需对DBE(数据库引擎)的支持。使用内存数组作缓冲工作区创建图表可选用TChart构件，图表的数据来自内存数组，所以它实际上已不是一个数据库图表。为了配合TChart图表的动态显示，内存数组的设置应与图表显示区一样大，同时需要给数组设置一个动态指针。指针的操作由图表的动态显示形式决定。如果图表上显示的动态曲线一走完显示区就后退一个时间段，然后继续向前走(大多数工业图表都采用这种显示形式)，那么指针的对应操作应是：数组按指针指定位置每次接收一个新数据，指针就加1，然后判断指针相对于数组是否越界，若越界，则指针减去一个时间段的对应值，相应地数组元素随指针依次前移。由于图表的数据来自数组，所以图表构件相应地要使用Series —>Addy 和 Series —>Delete 方法来添加及删除对应的数据。

3　应用实例

我们将应用实例的程序结构设计为通用的多窗口模式，即由主窗体和一系列子窗体构成。主(窗体)程序负责数据采集 b 处理及保存数据到磁盘数据表，并设置一系列按钮(或菜单)，如图1，通过触发这些按钮的事件响应函数，调出子窗体显示这些过程参数的动态曲线图表。

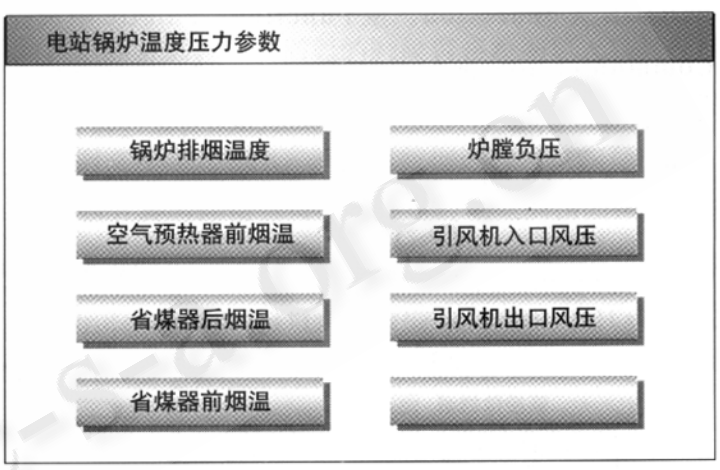


图 1　应用实例主窗体

3.1 工作原理

设计数组作为动态曲线图表的缓冲工作区，当不显示子窗体图表时，采集的数据仅送到数组缓冲区(后台作业运行方式)，当显示图表时，数组缓冲区的数据立即再现到图表上。

大多数变化慢的工业数据没有必要保存每秒的采集值，通过设置一个30秒定时器，可以每隔30秒进行一次平均值计算，然后将平均值存入磁盘数据表，这样，大大减少了访问磁盘的次数，提高了程序的运行水平。

3.2 设置构件属性

在子窗体（这里以Form2为例）的Tchart构件上，单击鼠标右键，选择弹出菜单的Edit Chart命令，即进入图

表编辑器。首先在图表上增加两个TLineSeries构件，分别作为锅炉甲侧和乙侧排烟温度的显示曲线。在Chart项Paging页的Points Per Page框内输入每页点数，如果定义图表显示区为10分钟时间段，则输入600。Scale Last Page框不选（即置为false），以保证图表具有固定的显示区。因为工业数据的量程范围是确定的，所以，我们把图表Y轴的上下限人为输入定值，进入Chart | Axis | Scales子项，在左边的Axis框指定Left（Y轴），不选Automaitic（即置为false），而人为输入Mininium值为0，输入Maximum值为200。X轴的范围不人为确定，设置由系统自动调整，在Chart | Axis | Scales子项，先在左边的Axis框指定Bottom（X轴），选择Automaitic。注意，X轴的范围必须设定为自动，否则，当程序删除TLineSeries序列前面的旧数据时，曲线不会自动后移，还需要你手工编程来处理曲线的后移。

另外Chart构件虽然提供将X轴或Y轴定义为DateTime的属性设置，但并不好用。因为它实际上是把X轴设置为日期轴，对于我们并不适用，所以不需要设置它。这样，X轴仍作为一般的数据轴，它的刻度按照Points Per Page框的输入值显示，并不代表时间。为了指示当前时间，我们在Chart构件下面加一个Panel构件，将它的Caption属性赋值为Time()函数，Align属性置为atBottom。

至此，主要属性参数设置完毕，其他属性参数可沿用构件的缺省值。

3.3 主程序清单：Unit1.cpp(略)

3.4 应用程序的优化

当窗体较少时，上述程序完全可以正常工作。如果进行大型应用程序设计，由于运行时系统要为众多窗体配置内存空间，系统资源很可能被用尽。这时程序运行将变得十分缓慢，或者根本无法运行，系统也会发出内存不够的通知，请你关闭一些程序再运行。所以，合理的设计方法应该是采用动态方式建立窗体，即在程序运行过程中用到哪个窗体才建立，用完后马上删除并释放所占用内存资源。例如，当按下Button1时，才动态地建立Form2窗体：

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
  Form2 = new TForm2(Application);
Form2->Show();
}
```

当按下Form2窗体的关闭按钮Button1时，即关闭窗体并删除Form2。在Form2的单元文件Unit2.cpp中，它是通过以下两个事件处理函数实现的：

```
void __fastcall TForm2::Button1Click(TObject *Sender)
// 关闭窗体 Form2
{
 Close();
}
void __fastcall TForm2::FormClose(TObject *Sender, TCloseAction &Action)
// 这是 Form2 的 OnClose 事件函数。对于用运算符 new 建立的动态对象，一般使用运算符 //del
```

ete来释放所分配的内存。但关闭窗体事件函数包含一个参数Action，它有四种状态：//caNone、caHide、caFree、caMinimize，其中caFree专用于释放窗体占用的内存，所以在//此应使用参数Action的caFree状态来释放内存。如果在这里使用语句delete Form2，会出//现‘Abstract Error’的错误信息。

```
{
 Action = caFree;
}
```

对应于窗体的动态建立方式，C++Builder的设置也需要做一些相应的改变。选择Project | Options打开工程选项窗口，再选Form页，在Main form框选定Form1作为主窗体，然后在Auto-create form子窗口中，把Form2、Form3...全部移到Available form子窗口，即设定这些窗体不由系统自动建立，而是置于(尚未建立的)可用状态。

细心的读者又会发现，这样的程序在运行时，同一按钮可以重复地打开同样的窗口，这是不必要的，容易引起混淆。对此，有些书籍为读者编写了一个专门的函数，用于判断窗体是否已经存在于屏幕上，调用这个函数来禁止窗口的重复打开。然而，能否更简单地利用Form构件自身的特征来处理这个问题呢？

我们注意到，Form实际上是一个窗体指针。动态创建窗体时，使用new语句为窗体对象分配内存，并返回一个指针给Form，指针指向动态分配给窗体的内存空间。最初程序给每个窗体定义指针时，这些窗体并不存

(下转第78页)

(上接第 76 页)

在，所以Form的初值为NULL。窗体需要使用时就被建立，同时系统将指向该窗体内存空间的地址值返回给Form。删除窗体时，使用delete语句释放其所占用的内存。下次再用到该窗体时，Form又被赋予一个新的内存地址值。注意，指针这种重复赋值的过程都是发生在窗体建立时，而在删除窗体时，delete语句仅释放内存并不给Form返回值.(也不需要返回值)。所以，我们只要在delete语句（或者Action = caFree语句）之后，人为地添加一个为Form赋初值(NULL)的命令，就可以利用Form之值来判断窗体是否存在，关于重复打开窗体的问题以及其他有关窗体存在的问题也就迎刃而解了。

为了禁止重复打开同一窗体，将Button1的OnClick事件函数改写为：

```
void __fastcall TForm1::Button1Click(TObject
*Sender)
  {
  // 如果窗体不存在，建立该窗体
   if (Form2 == NULL)
    {
     Form2 = new TForm2(Application);
     Form2->Show();
      :
    }
```

将Form2的单元文件Unit2.cpp中Button1的OnClick事件函数改写为：

```
void __fastcall TForm2::FormClose(TObject *Sender,
TCloseAction &Action)
  {
   Action = caFree;
   Form2 = NULL;
  }
```

最后，在定时器 1 的事件函数中，还应将两处的 if (Form2->Visible)改为 if (!Form2 == NULL)。■

参考文献

1 陈周造，陈灿煌 编著，C++ Builder4 彻底研究，中国铁道出版社，2000.1。

2 王有一，创建实时数据库动态图表，计算机系统应用，2001.2。

3 http://www.inprise.com/bcppbuilder/

4 杜民 主编，陶成钢、祝朝辉、李讫波、刘潞、顾旗君 编，C++ Builder4.0程序设计与开发指南，高等教育出版社，1999.12。