

C++ Builder 中 超文本结构的实现

湖南医科大学基础医学院 李晓春 元秀华 李玲

介绍了 C++ Builder 中超文本的制作方法, 分析了编程难点, 并针对难点给出解决办法。

超文本的实现方法与存在的问题

1. 超级链接

实现超文本结构最简单实用的办法是采用超级链接。C++Builder(以下简称 CB)支持对象的嵌入和链接技术, 在它的元件板的 System 页上有 OLE 控件。若主文本框选用 TRichEdit 控件, 可在文本框中静态地创建容器服务器 OLE 控件, 双击 OLE 控件进入“插入对象”对话框, 选择插入的对象文件及对象在主文本中显示的图标式样, 将对象链接到文本当中。若 OLE 控件的 Enabled 属性设为 true, 程序运行时容器可见, 双击对象图标就可完成对象媒体的调用。若 Enabled 属性设为 false, 程序运行时容器不可见, 这时可通过其他控件(如菜单条 N1)来实现调用:

```
void __fastcall TForm1::N1Click(TObject *Sender)
{
    OleContainer1->DoVerb(0);
}
```

这种方式虽然简单、灵活、具有广泛的适应性, 但千篇一律的调用方式过于单调, 缺乏想象力。而且对象总是在原编辑环境中打开, 不同对象打开时的界面各不相同, 缺乏统一感。

2. “热字”、按钮与图片

目前采用较多的是在文本框中设“热字”、按钮、图片等控件来实现媒体间的连接 [2], 这极大地增加了媒体的表现力。在主文本框中, 只要轻点“热字”、按钮、图片等控件, 就可以再打开文本、图形或影音文件, 完成超媒体功能。在 CB 中可以用作链接媒体的控件较多, 如“热字”可以是 TRichEdit、TStaticText、TEdit、TMemo 等的对象, 标准按钮 TButton 上的文字也可以设定为热字, 用来连接其他的媒体。按钮控件可选标准按钮 TButton 或图形按钮 TBitBtn, TBitBtn 的 Glyph 属性可以让你装入想要展示的图片。作为图片控件只要选择 TImage 即可, 或者就用图形按钮 TBitBtn。这些控件都有 OnClick

事件, 大部分控件还有 OnDblClick 事件, 对这些事件进行编程就可以实现超媒体的调用。如以 StaticText 的文字作为“热字”, 为了使“热字”醒目, 又不破坏主文本的格式。可设定热字无边框, 颜色为红色(一般文本框内字的颜色缺省为白底黑字), 即:

```
StaticText->Font->Color=clRed;
```

```
StaticText->BordorStyl=sbsNone;
```

字体及大小设定为与主文本框字的特性一样, 并将 StaticText 放置于主文本框中恰当的位置。字颜色为红色, 表示为热字, 一目了然。假设热字连接的事件是显示图片和播放解说文件, 用鼠标单击来触发该事件, 则在代码编辑器中编写代码:

```
void __fastcall TForm1::StaticText1Click(TObject *Sender)
{
    Image1->Picture->LoadFromFile("P1.bmp");
    Image1->Stretch=true;
    Image1->Visible=true;
    MediaPlayer1->DeviceType=dtAutoSelect;
    MediaPlayer1->FileName="P1.wav"); //P1.wav
    为图片 P1.bmp 的解说文件
    MediaPlayer1->Open();
    MediaPlayer1->Play();
}
```

双击 Image1 中的图片, 停止解说并退出图片:

```
void __fastcall TForm1::Image1DblClick(TObject *Sender)
{
    Image1->Visible=false;
    MediaPlayer1->Close();
}
```

利用这种方法可以实现在文本框中调用其他任意的

媒体,即超媒体的调用。但作为连接其他媒体的控件,它们的位置相对于文本应该是不变的。若文本移动,媒体控件也应该同步移动。在CB中,文本中的“热字”、按钮、图片都是“浮”在文本上的,若文本框较小,不足以显示整个文本时,则文本框中会出现针对文本的垂直滚动条。该滚动条只能滚动文本,而不能滚动媒体控件。垂直移动滑块时会造成文本与控件之间的错乱。若将文本分成多页显示,则既浪费系统资源,又使文本缺乏连贯性。同时,由于各控件的位置在空间上总是与文本框部分重叠,若控件“浮”在上层,控件是可见的,若某控件是“沉”在文本框下,就会因不可见而不可操作。这些都是超文本设计的难题。

解决方法

1. 文本和控件的同步移动

实现文本和控件的同步移动有两种办法,其一是保存文本框的长度和宽度,以及各控件的位置为全局变量,并根据滑块的移动位置相应地调整各控件的位置。这需要作些计算,较为麻烦。另一种较为简单的办法是将文本框TRichEdit的宽度设计为与窗宽一致,而将文本框的高度(Height属性)设计得足够大,能容纳整个要展示的文本。这样一旦文本框超过窗口的高度,窗口内会出现垂直滚动条。这种由于窗体内控件超过窗体而出现的属于窗体的滚动条对窗体内的所有控件都起作用。当文本框移动时,框内设计好的“热字”、按钮、图片等控件和文本就会始终保持同步运动。

2. 控件的可见性

只有在主文本框中可见的控件才可以进行操作。在RichEdit中,其他的文本控件TRichEdit、TStaticText、TEdit、TMemo都是可见的,但作为“热字”的字的个数不会太多,且不需编辑,以选用TStaticText为佳,其属性控制也较其他控件容易一些。标准按钮在RichEdit中也是可见的,但标准按钮突起于文本框,若非特别需要,一般不用。图象控件无论从其形式上,还是从其表达的内容上看都是比较合适的连接控件,但TImage控件在TRichEdit中为不可见。图形按钮虽然可见,但其图象控制极为不便。因此需要其他的载体来显示图形。CB中的TPanel控件是一个用途十分广泛的控件,它具有窗体的特性。它可以把窗口分为几个部分,也可以作为其他元素的容器。利用它作为图象的容器十分方便,并且在TRichEdit中可见。为此,在主文本框需要放置图片的地方先放置Panel控件,用鼠标拖放,使其面积大小为图象

控件的尺寸。再选择TImage控件置于其上,TImage的Align属性设为alClient,图象框充满整个TPanel的大小。TPanel在TRichEdit中可见,而TImage覆盖在TPanel上,因而也可见。这种方式下,图象连接的媒体应该由TImage相关的事件来控制。而图象控件是否可见则由TPanel的Visible属性来决定。若单击图象控件事件设计为动态地创建另一窗口(Form2),并显示其中的图象和文本,则在创建工程文件时,先设计好窗体Form2及其中的图象和文本等内容,选择CB开发环境中Project菜单的Options选项,进入Project Options对话框,将Form2设定为非自动创建窗口。这样在程序运行时Form2窗体并不创建,Form2窗体的创建由图象控件的单击事件来触发,编程如下:

```
void __fastcall TForm1::Image1Click(TObject
*Sender)
{
    TForm2 *Form2=new TForm2 (this);
    Form2->Show();
}
```

这样,由图象控件就可以方便地、动态地创建其他媒体视图或其他窗体。不需要窗体时,关闭窗体,并释放内存 [3]:

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

总之,超文本的媒体组织方式越来越受到人们的欢迎。C++ Builder中隐藏了Windows编程的复杂性,使对媒体的操作更加易于控制,能方便地实现超文本结构。作者用C++ Builder3.0制作了“生物信号数字处理多媒体实验系统”,在其中采用了上述的超文本结构方法,灵活、方便地将各种媒体组织在一起。系统操作自由、简单,充分体现了Windows应用系统的多任务、互动的特点。■

参考文献

- 1 徐新华编著. C++ Builder3 编程指南. 北京:人民邮电出版社, 1998年
- 2 刘南迎. VFPCA1 “热区”与超文本、超图象技术的实现. 计算机应用 1999; 4: 47-49
- 3 许冬岗等编著. Borland C++ Builder 编程技巧与实例. 西安:西安电子科技大学出版社, 1998年