

基于TreeView和数据库的多媒体 课件导航技术

Navigation Technology in Multimedia Courseware Based on TreeView and Database

范士娟 杨超 (南昌华东交通大学机电工程学院 330013)

摘要: TreeView是Windows应用程序中功能强大而又使用广泛的控件,将该控件与数据库结合起来用于多媒体课件的导航可给用户带来很大方便。介绍了Delphi中TreeView控件的常用属性和方法,举例说明了在多媒体课件中使用该控件树状显示、修改数据库数据的基本方法,以及根据选中的条目定位对应的媒体文件,并根据文件类型使用相应的播放控件显示/播放文件,从而实现其导航功能的方法。

关键词: TreeView 控件 导航 多媒体课件 数据库 Delphi

在Windows资源管理器、许多应用软件以及某些管理系统软件[1]的界面上,常常有一个树视图(TreeView)窗口,里面显示的是相应内容的分级结构表,点击该窗口中相应条目即可查看相应的内容,用户甚至可以在该窗口中对分级结构表进行修改(移动、添加、删除等)。数据库是用电子计算机对资料进行管理、实现资料共享的一门技术,利用数据库对多媒体课件内容进行管理,有助于用户按照自己的意愿安排内容。在多媒体课件中结合TreeView和数据库进行导航,可给用户带来很大方便。本文着重介绍在多媒体课件中的利用该控件和数据库实现导航的方法。

1 TreeView 控件简介

在Delphi环境里TreeView控件是一个显示文档标题、索引或磁盘文件和目录等项目分级表的窗口,它显示按照树形结构进行组织的数据,广泛地用于实际中,它的结构层次虽然比较复杂,但具有强大的功能。TreeView控件中的一个树形图由节点(TreeNode)和连接线组成,每个节点又包含标签文本、数据和若干个位图图标;每个节点还可包含若干个子节点,用户可以展开或折叠父节点下的所有子节点。TreeNode对象(节点)是TreeView控件的基本组成单元,TreeNode对象是TreeNode对象的集合,TreeNode对象构成TreeNode对象的item属性。要想很好地使用TreeView控件完成导航功能,除了要熟悉该控件的结构层次关系外,还必须熟悉该控件常用的属性和方法,下面简单

介绍TreeView控件部分常用的属性和方法。

1.1 TreeView 控件常用的属性与方法

掌握该控件常用的属性与方法,有助于很好地使用该控件。

(1) AutoExpand属性: 设为True, 则被选择的条目自动展开, 未被选择的条目自动折叠;

(2) DragMode属性: 拖动模式, 可设为dmManual(手动)或dmAutomatic(自动);

(3) DropTarget属性: 节点在拖动中是源还是目标。

(4) Items属性: TreeNodes对象, 包含TreeView控件中所有显示的节点;

(5) Selected属性: TreeView控件中被选中的条目;

(6) LoadFromFile方法: 从文本文件读入数据并装入TreeView控件, 与SaveToFile方法配合使用;

(7) SaveToFile方法: 将TreeView控件的条目数据保存为文本文件, 与LoadFromFile方法配合使用;

1.2 Tree Nodes 对象常用的属性与方法

TreeNode对象是TreeView控件的Items属性。节点的插入、删除等操作, 以及将文件或数据库里的数据装入到TreeView控件都是通过该对象的属性和方法来实现的。

(1) Count属性: TreeNode对象包含的节点个数;

(2) Item属性: TreeNode对象管理的所有节点;

(3) Add、AddFirst、Insert方法: 在同一级别的最后位置、开

始位置、指定位置上增加或插入一个节点；

(4) AddChild、AddChildFirst方法：在当前节点下一级别的位置、开始位置增加一个节点；

(5) Clear方法：删除TreeNode对象管理的所有节点；

(6) Delete方法：删除指定的节点；

1.3 TreeNode 对象常用的属性与方法

TreeNode对象是TreeNode对象的Item属性。对文件、数据库里的数据、TreeView控件的数据结构进行修改，以及利用TreeView控件实现导航功能，都是通过该对象的属性和方法来完成的，掌握该控件常用的属性与方法，有助于完成数据的修改和导航工作。

(1) AbsoluteIndex属性：表示从TreeView第一个条目开始的顺序值，第一个是0，第二个是1，如此类推；

(2) Count属性：属于某节点的所有子节点的个数；

(3) Expanded属性：是否展开所有的子节点；

(4) HasChildren属性：某节点是否有子节点；

(5) Index属性：某节点在同一级别所有节点中的索引，第一个是0，第二个是1，依次类推；

(6) Item属性：当前节点的某个子节点；

(7) Level属性：当前节点在TreeView控件中的级别；

(8) Parent属性：当前节点的父节点；

(9) Text属性：识别节点的标签；

(10) Delete方法：删除节点及其所有子节点；

(11) DeleteChildren方法：删除节点的所有子节点；

(12) GetFirstChild、GetPrevChild、GetNextChild、GetLastChild

方法：返回当前节点的第一个、前一个、下一个和最后一个子节点；

(13) IndexOf方法：返回节点在同一级别里的序号；

(14) MoveTo方法：将节点由一个位置移动到另一个位置；

2 建立数据表

在多媒体课件中，为了使用户能够对授课内容信息进行修改、调整或删除不需要的信息、添加新的信息，使用数据库是比较方便的方法。在数据库里可设计若干个数据表，用来存放多媒体课件内容所指定文件的位置等信息。例如：设计授课内容表(TeachTable.dbf)存放授课内容文件的有关信息，该数据表可设计4个字段：Z[章节]、MS[描述]、WJ[M文件名]、SY[索引]，字段数据类型全部采用字符型，前3个字段的长度可根据需要确定。“Z”用来指示多媒体课件内容所属章节，一般作为父节点的显示文本；“MS”用来存放子节点的显示文本；“WJ[M”用来存放子节点对应的多媒体课件内容的媒体文件名；“SY”的长度为4，前两位表示章序号(01-99)，同时也表示该条目的父节点在TreeView控件中的位置，后两位表示节序号(01-99)，同时

也表示该条目在其父节点包含的所有子节点中的位置，只要改变该字段值就可改变记录在TreeView控件中的位置。

3 多媒体课件中的导航技术

在多媒体课件中将TreeView控件与数据库联系起来，能将数据库里的数据(课件内容信息)以层次结构的形式清晰地显示出来；利用该控件可以方便地对数据进行增加、修改、删除等操作，借助该控件的拖放技术还可实现数据的移动，从而实现内容的调整；选中TreeView控件里树形图的节点并根据该节点的信息定位到数据库的相应记录，即可实现多媒体课件的导航功能，给用户带来方便。

3.1 将数据库数据装入TreeView 控件

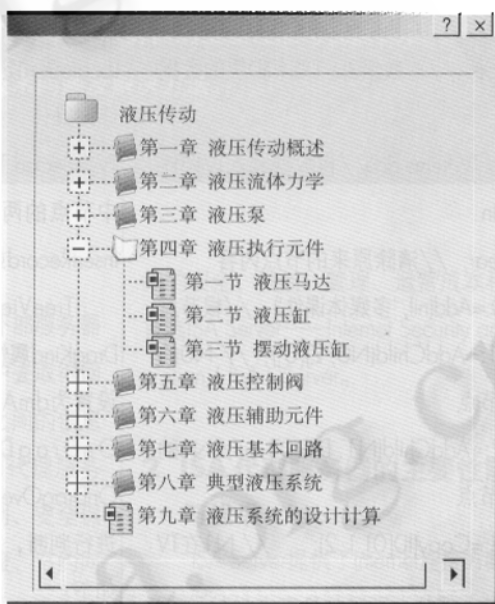


图 1

在导航窗体上放置TreeView控件TV、数据表控件Table1并进行必要的设置，使Table1与TeachTable.dbf相连接。下面的方法将Table1里的数据装入TV：首先定义三个数组Z、DIS和ID，分别存放Table1里字段“Z”、“MS”和“SY”的值，然后对ID里的数据按从小到大的顺序排序，Z和DIS里对应的数据也同时调整顺序，最后根据ID里的数据利用循环将Z和DIS里的数据添加到TV里，如图1所示。实现过程如下：

```
var N,I: Integer;
    N0,N1,N2: TTreeNode;
    Ind: String[2];
begin
    Table1.Open;
    N:= Table1.RecordCount;
    SetLength(Z,N); SetLength(DIS,N); SetLength(ID,N);
```

```

Table1.First; l:=0;
With Table1 do
Begin
Repeat //将数据表里的数据装入数组
Z[l]:=Trim(FieldByName('Z').AsString);
DIS[l]:=Trim(FieldByName('MS').
AsString);
ID[l]:=Trim(FieldByName('SY').
AsString);
l:=l+1; Next;
Until Eof;
end;
SetSort; //对数组Z、DIS和ID里的数据
排序
With TV.Items do //将数据装入TV控件
begin
Clear; //清除原来的节点内容
NO:=AddNil, '多媒体课件'; //根节点
N1:=AddChild(NO, Z[0]); // NO的
第一个子节点
N2:=AddChild(N1, DIS[0]); // N1的
第一个子节点
Ind:=CopyID[0], 1, 2); // N1在TV
中的索引
for l:=1 to High(ID) do
begin
if CopyID[l], 1, 2)=Ind then //父索
引相同
N2:=AddChild(N1, DIS[l]); // N1
的子节点
Else begin //父索引不同
N1:=AddChild(NO, Z[l]);
Ind:=CopyID[l], 1, 2);
N2:=AddChild(N1, DIS[l]); // N1
的子节点
end;
end;
end;
end;
end;

```

3.2 增加、删除、编辑、移动树节点及数据库数据

TV里的条目可以通过修改Table1里的数据后再将Table1的数据重新装进TV里,也可以在TV里直接修改。直接在TV上增加、删除、编辑节点只需用它本身提供的TV.Items.AddChild、TV.Selected.Delete(注意:对于子节点只删除子节点本身,而对于父节点将同时删除其包含的所有子节点)、TV.Selected.EditText等方法,同时要相应修改数据库中的数据;删除和编辑节点操作,可根据选中条目的标签文本或节点的两个索引用Table1的Locate语句定位到相应的记录,然后作相应的修改操作;如定位语句:Table1.Locate('MS', TV.Selected.Text, []).增加节点操作也是根据选中节点的两个索引使用Table1的Insert、InsertRecord或Append方法插入或添加记录。

TreeView控件支持拖动操作,将组件的DragKind属性设置为dkDrag, DragMode属性设置为dmAutomatic,并为OnDragOver与OnDragDrop事件编写处理程序即可。OnDragOver事件处理程序对允许移动的条件进行判断,排除需要避免的特殊情况,代码如下:

```

var
TargetN, SourceN: TTreeNode;
begin
TargetN:= TV.GetNodeAt(x, y);
If {Source=Sender} and {TargetN <> nil}
then
//保证移动在TV上,目标节点不为空
begin
Accept:=true;
SourceN:= TV.Selected;
While {TargetN.Parent<>nil} and
{TargetN <> SourceN} do
TargetN:= TargetN.Parent;
If {TargetN = SourceN} then Accept:
=false;

```

```

End else Accept:=false;
If Accept then
TV.Selected.MoveTo(TargetN, naInsert);
end;
在OnDragDrop事件里编写处理程序以修
改数据表里的数据,可根据移动后条目的标
签文本或节点的两个索引用Table1的Locate语
句定位到相应的记录,然后修改“ZJ”和
“SY”字段值。例如:
Var Ind1, Ind2:String[2];
I1, I2:Integer;
With Table1 do
Begin
Locate('MS', TV.Selected.Text, []);
I1:= TV.Selected.Parent.Index; I2:= TV.
Selected.Index;
If I1<9 then Ind1:='0'+IntToStr(I1+1) else
Ind1:= IntToStr(I1+1); // I1表示所选节点父
索引
If I2<9 then Ind2:='0'+IntToStr(I2+1) else
Ind2:= IntToStr(I2+1); // I2表示选节点索引
FieldByName('ZJ').AsString:=TV.Selected.
Parent.Text;
FieldByName('SY').AsString:=Ind1+Ind2;
End;

```

3.3 实现导航

当点击TreeView控件里的条目时,可根据条目的描述文本和/或索引定位到数据表里的相应记录,取出“WJM”字段对应的文件,根据文件类型使用相应的播放控件显示/播放该文件。例如:

```

var File_Name, File_Ext: String;
begin
With Table1 do
Begin
Locate('MS', TV.Selected.Text, []);
File_Name:=FieldByName('WJM').
AsString;

```

```

File_Ext:= ExtractFileExt(File_Name); /
/文件扩展名
If File_Ext ='.avi' then
Begin
    MediaPlayer1.FileName:
    =File_Name;
    MediaPlayer1.Play;
End Else If File_Ext='.swf' then
...
End;
End;
```

4 结束语

TreeView控件的结构层次虽然比较复杂,方法、属性较多,但功能很强,在多媒体课件中将该控件与数据库联系起来,利用该控件显示、修改数据库里的数据并对内容进行导航,能够给用户带来很大的方便。本文举例说明了使用该控件树状显示、修改数据库数据的基本方法,介绍了如何根据选中的条目在数据表里找到对应的媒体文件名,并根据文件类型使用相应的播放控件显示/播放文件,从而实现其导航功能的方法。本文只介

绍了两级结构的导航技术,多级导航技术要复杂些,可在此基础上扩展;在作者开发的《液压传动》CAI集成演示平台里就利用了本文介绍的导航技术。