辅助决策应用程序中决策栅格数据的动态报表

黄晓微 张仁平 卜雅原 (重庆后勤工程学院自动化工程系 400016)
张建 (重庆创新生物工程有限公司 400016)

摘要：本文主要结合实际工程应用，给出了用 Delphi 开发决策支持应用程序中决策栅格控件 TDecisionGrid 动态报表的解决方案，并结合解决方案，说明了实现中应该注意的事项。

关键词：决策支持 多维数据 动态数组 动态报表

1 引言

Delphi 作为强大的数据库开发工具，正被愈来愈多的编程人员所采用，“聪明的程序员用 Delphi”更形象生动的道出广大程序员的心声。在 Delphi 4.X 以上版本中提供了一组用于决策支持的控件，该组控件及其作用见表 1。使用该组标准控件非常方便地引入数据集，建立数据仓库，用户可以通过数据透视表上的按扭选择任意维数和某个统计项的统计值对数据进行全方位、多角度地分析。然而，如何将用户在决策栅格中组织的动态分析数据报表输出？显然没有现成的解决方案，因为决策栅格中的分析数据是随用户不同的操作意图不断变化而变化。下面就这个问题提出一种解决方案，实现决策栅格数据的动态报表输出。

表 1 决策支持的控件及其作用表

TDecisionCube	相当于多维的数据仓库
TDecisionQuer	用于引入数据集，建立数据仓库
TDecisionSource	与 TDataSource 相似，为下面的三个决策控件提供数据源
TDecisionGrid	以栅格形式显示用户分析组合的多维数据
T DecisionPivot	用于对栅格和图表重新组合
T DecisionGraph	以图表形式显示用户分析组合的多维数据

2 决策栅格动态报表的解决方案

下面，在某公司人事管理系统的数据分析子系统中，结合辅助决策应用程序(运行界面如图 1 所示)说明决策栅格数据实现动态报表的方法和步骤。

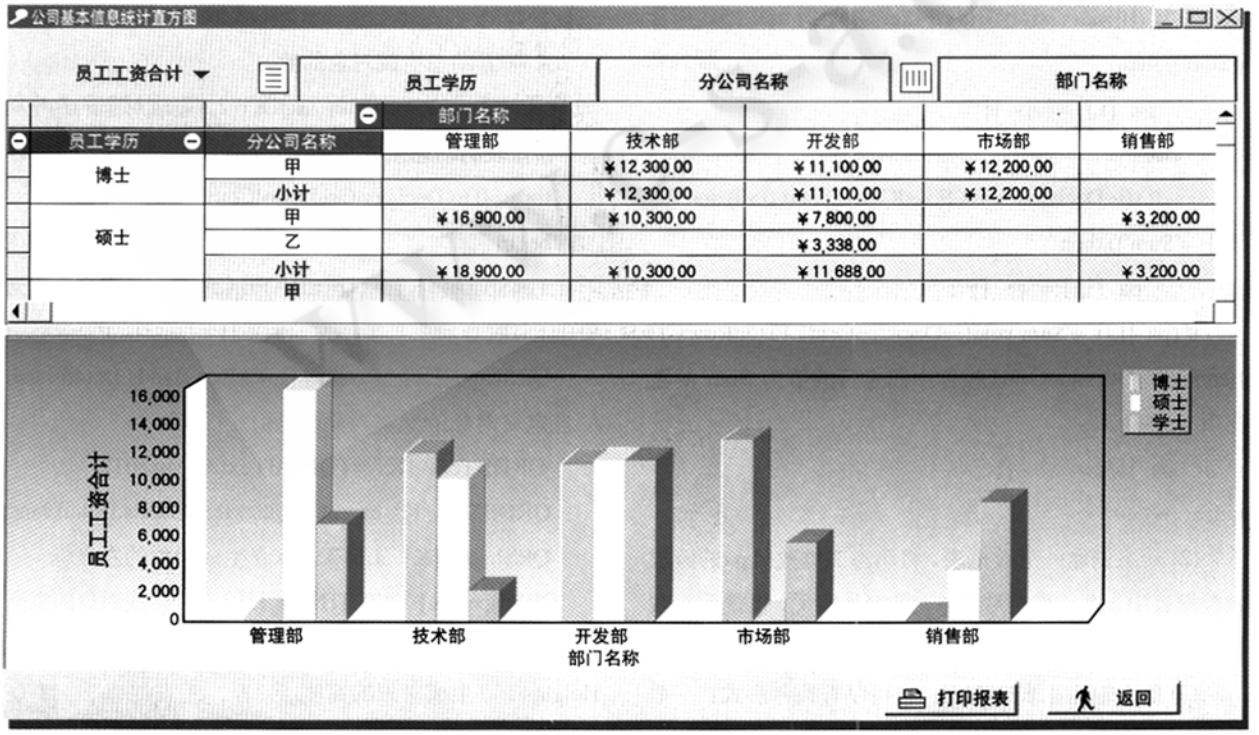

图 1 数据分析子系统的辅助决策应用程序运行图

笼统地讲，可将决策栅格数据动态报表的实现步骤分解如下：首先从控件 TDecisionGrid 中获得决策栅格数据，放到动态设置的二维数组 pa 中；其次，动态创建一个数据表存放报表数据；最后将动态创建的数据表动态报表输出。具体而言，以上步骤和方法细化如下：

(1) 根据 TDecisionGrid 控件的 cells 属性，获得报表所需数据，并将其默认的'Sum'值汉化成'总计'、'合计'、'小计'以符合汉语的习惯要求，所求得的动态分析数据存放于二维数组 pa 中，pa 的行和列完全由动态分析产生的数据来确定，这是第一步，也是最重要的一步。动态设置二维数组 pa，其标准定义式为 pa:array of array of string，动态改变二维数组 pa 的行和列的语句是：SetLength (pa,m,n)，要获得动态分析的数据和经汉化的'Sum'值可通过以下方法：

```
for i:=0 to DecisionGrid1.ColCount-2 do
    for j:=1 to DecisionGrid1.RowCount-1 do
    begin
      pa [i,j]:=DecisionGrid1.cells [i-DecisionGrid1.FixedCols+1,j-DecisionGrid1.FixedRows];
    if ((i=0) and (pa[i,j]='Sum')) then//处理DecisionGrid1控件中固定列的值为 'Sum' 的数据项
      pa [i,j]:='总 计'
    else
      if ((i =DecisionGrid1.FixedCols-2) and (pa [i,j] ='Sum')) then
        pa [i,j]:='小 计'
      else
        if ((i<DecisionGrid1.FixedCols-2) and (i>0) and (pa [i,j] ='Sum')) then
          pa [i,j]:='合 计';
    if (pa [i,j] ='Sum')and(j=DecisionGrid1.FixedRows-1) then// 处理 DecisionGrid1 控件中固定行的值为 'Sum' 的数据项
      pa [i,j]:='总 计';
end;
```

(2) 动态创建一个数据表，将动态二维数组pa的值放入数据表中，这一步相对简单，较容易实现，也就无须我赘述了。

(3) 确定输出报表的宽度。它可以有两种形式：一是数字区(cells 值只是数字的那部分)是等宽，即报表数字区的宽度完全由最大数据项的长度来确定；二是数字区的宽度由该列的最大数据项的长度来确定。最后根据确定后的报表宽度(含非数字区)来确定报表超宽？横向报表还是纵向报表？

(4) 在动态报表前取消系统对报表控件 TQuickRep的控制，以避免再一次对动态分析产生的数据报表输出时出现错误。例如对 TQuickRep 的 DetailBand. 上所控制的所有控件进行取消控制，其实现的方法如下：

```
for i:=1 to QuickRep.Bands.DetailBand.ControlCount DO
  QuickRep. Bands. DetailBand. RemoveControl (QuickRep.Bands.DetailBand.Controls [0]);
```

(5) 报表的动态生成。这是所有工作中最困难的一步，在该步实现的方法中，要注意每一个动态生成的 TQRShape 控件和 TQRDBText 控件的高度和宽度，以及它们的 Left 属性值(即对应于左坐标)和 Top 属性值(即对应于上坐标)。若在报表的不同部分，如 TitleBand 和 DetailBand 中的 TQRShape 控件，要分别创建和指定其相应的父类对象 TitleBand 和 DetailBand。另外，报表的标题可加上 TDecisionGraph 控件的左坐标名，以区别于不同的分析项目。具体实现可用以下方法：

```
SetLength(QRShape,DecisionGrid1.ColCount); // 动态设置数组
SetLength(QRDBText, DecisionGrid1.ColCount); // 动态设置数组
K:=0;// 动态生成对象的数，
Lx:=(QuickRep.Width-ZK)DIV 2;//生成对象的左坐标
// 报表的动态生成
for i:=0 to DecisionGrid1.ColCount-2 do
begin
  QRShape [K]:=TQRSHAPE.Create(tj1);// 自定义对象的创建(下同)
  QRShape [K] .Parent:=QuickRep.Bands.DetailBand; // 自定义对象的父类对象(下同)
  QRDBText [K]:=TQRDBText.Create(tj1);
  QRDBText[K].parent :=QuickRep.Bands.DetailBand;
  QRShape [K] .LEFT:=Lx;// 生成对象的左坐标
  QRShape [K] .WIDTH:=M [J] +2; // 生成对象的宽度
  QRShape[K].HEIGHT:=QuickRep.Bands.DetailBand.Height+1; // 生成对象的高度
  QRShape [K] .TOP:=-1; // 生成对象的纵坐标
  QRDBText [K] .WIDTH:=QRShape [K] .WIDTH-10;
```

```
QRDBText [K] .Left :=QRShape [K] .LEFT+1;
QRDBText [K] .HEIGHT:=QRShape [K] .Height div 2;
QRDBText [K] .Top :=QRDBText [K] .Height div 2+QRShape [K] .Top;
QRDBText [K] .AutoSize:=false;
QRDBText [K] .Alignment:=taCenter; //生成对象居中
QRDBText [K] .DataSet:=Table1;
QRDBText [K] .DataField:=IntToStr(k+1);
Lx:=Lx+M [J] +1;
Inc(k);
end;
//动态生成报表的标题
Caption := TQRLabel.Create(tj1);
Caption.Parent := QuickRep.Bands.TitleBand;
Caption.Alignment:=taCenter;//标题居中
Caption.Width:= Length (Trim (ptitle))*8; //标题的宽度
Caption.Left:=(QuickRep.Width- Length (Trim (ptitle))*8) div 2; //标题的左坐标
Caption.Height:=QuickRep.Bands.TitleBand.Height-1; //标题的高度
Caption.Top:= 0;
Caption.Caption:=ptitle; //标题的名称，ptitle 为调用该公用模块时的输入参数
QuickRep.DataSet:=Table1;
QuickRep.Preview;
```

(6) 报表结束时，不要忘记释放动态生成的对象并关闭动态产生的数据表。

3 应注意的问题

(1) 在 TQuickRep 控件相应出现的 HasColumnHeader、HasDetail、HasTitle 三个属性必须设置为 true；

(2) 不能忘记 TQuickRep 控件与动态产生的数据表相连，即源代码中的 QuickRep.DataSet:=Table1 语句；

(3) 若对报表数据的字体、大小等属性值进行修改，则动态生成组件的宽度计算必须放在其后来完成；

(4) 另外，动态数组给定的内存(即数组容量)以及指定动态数组的起始位置(不一定为 0，根据 DecisionGrid1 控件的固定列确定)很重要，因为一方面当数据库很大时它会大大减少内存的消耗，另一方面便于操作该数组，大大增强了程序的灵活性和通用性。

(5) 如果让 QRDBText 控件的数据居中，必须先设置其 AutoSize 属性为 false，然后才能设置其 Alignment 属性为 taCenter。这一点往往容易忽略，直接设置 Alignment 属性为 taCenter，往往达不到数据居中的目的。

4 结束语

以上详细介绍了决策栅格控件中的动态分析数据报表输出的解决方案，可将以上解决方案做成一个通用模块来实现，也可做成一个自定义组件或 ActiveX 控件。当然，由于客户对数据报表的可能特殊要求，此解决方案或许不能完全满足，但可以在此解决方案的基础上进行一些修改或补充，是可以满足大多数用户的要求的。例如，有些用户只关心数据表中的一个或几个字段值，则可以将动态产生的数据表的所有字段让用户进行选择，然后根据用户所选择的字段进行报表输出，而方案的其他部分几乎无须修改。■