第16卷 第5期
2006年5月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol. 16 No. 5
May 2006

用组合逻辑解决表单控件的控制问题

康仲奇[1],康 东[2]

(1.西安交通大学 电信学院,陕西 西安 710049;
2.咸阳师范学院 远程教育中心,陕西 咸阳 712000)

摘 要:因网页表单控件种类较多,故应用中交叉控制复杂。为使网页表单控件的控制简单和清楚,设计了如下工作方法:设定组合逻辑函数,简化其表达式,转化为伪码,程序语言表达。即用4个步骤和少许程序完成。这样,控制简捷,源码较少,步骤清晰,易于调试和修改。应用证明这种方法是可行和实用的。文中给出了一个抽取和精简的例子:text控件随着ridio控件激活,简要说明了这种方法的实施过程。

关键词:组合逻辑;控制;表单控件

中图分类号:TP311.1 **文献标识码**:A **文章编号**:1673-629X(2006)05-0155-02

Control Form Components by Combinational Logic

KANG Zhong-qi[1], KANG Dong[2]

(1. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;
2. Distance Learning Center, Xianyang Normal University, Xianyang 712000, China)

Abstract: The more kinds the form components have, the more complexity in applications their cross controls would be. For the simple and clear controls, have designed a method as follows: setting up some functions of combinational logic, simplifying expressions of them, translating the expressions into PDL and programing. Writed fewer source codes to control form components in four steps. In this way, the controls are simple and convenient, the codes are fewer, the steps are distinct, source language debugging and amending are easy. Some applications show the method to be feasible and practical. Give a selected and simplified example in this article: ridio component activates text component. By the example, intend to demonstrate how the method is carried out.

Key words: combinational logic; control; form components

0 引 言

由于网页表单控件种类较多,所以应用中交叉控制复杂。例如网页表单编程时,经常会遇到这样的问题:怎样将text控件随着ridio控件激活,并且源码较少?如图1所示。

这是所抽取、精简的一个例子。为了创建(Create)或修改(Modify)成批用户,有时需要从用户vc05142001到vc05142005依次递增进行,有时需要vb040820+01=vb04082001, vb040820+05=vb04082005…(+表示连接)分别选择进行。而text控件就要随着ridio控件成对或成组(两对)激活或蛰伏。由此可以抽象出组合逻辑问题逐步解决。

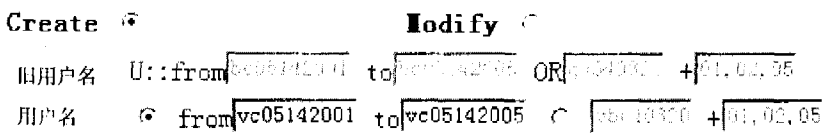

图1 text控件随着ridio控件激活

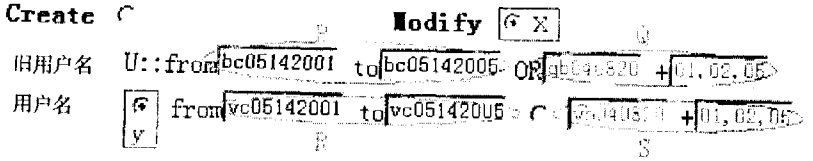

图2 设定逻辑变量和函数

1 解决方法

(1)这里不妨这样设定:

逻辑变量 x = Modify单选按钮选中,y = 用户名第一个单选按钮选中。

逻辑函数 P = 旧用户名第一对text控件激活,Q = 旧用户名第二对text控件激活,R = 用户名第一对text控件激活,S = 用户名第二对text控件激活。如图2所示。

(2)这样,有逻辑函数表达式:

$P = x \cap y$ $Q = x \cap \sim y$ $R = y$ $S = \sim$

收稿日期:2005-11-25
基金项目:国家高技术研究发展计划(八六三计划)(2003AA1Z2610)
作者简介:康仲奇(1954-),男,陕西人,高级工程师,研究方向为网络安全。

$y \quad \sim P = \sim (x \cap y) = \sim x \cup \sim y \qquad \sim Q = \sim (x \cap \sim y) = \sim x \cup y \qquad \sim R = \sim y \qquad \sim S = y$

其中，$\cap$ 表示逻辑与；$\cup$ 表示逻辑或；$\sim$ 表示逻辑非[1]。运算用到了德·摩根律[1]。复杂情况下，需要借助卡诺图[2,3]或其它方法简化逻辑函数表达式。

(3)用 PDL(类程序设计语言，伪码)表示[4]：

```
IF y THEN
  Enable ~Q,R,~S
  IF x THEN
    Enable P
ELSE
  Enable ~P,~R,S
  IF x THEN
    Enable Q
IF ~x THEN
  Enable ~P,~Q
```

(4)用 Java 脚本语言[5]：

```
function CoverUncover() //蛰伏或激活 text 控件
{if(document.edit.mode[0].checked)//y
  {document.edit.formers.disabled=1;document.edit.formersx.disabled=1;//~Q
  document.edit.user.disabled=0;document.edit.userx.disabled=0;//R
  document.edit.users.disabled=1; document.edit.usersx.disabled=1;//~S
  if(document.edit.method[1].checked) //x
    {document.edit.former.disabled=0;document.edit.formerx.disabled=0;}}//P
else{document.edit.former.disabled=1;document.edit.formerx.disabled=1;//~P
  document.edit.user.disabled=1;document.edit.userx.disabled=1;//~R
  document.edit.users.disabled=0;document.edit.usersx.disabled=0;//S
  if(document.edit.method[1].checked) //x
    {document.edit.formers.disabled=0;document.edit.formersx.disabled=0;}}//Q
if(! document.edit.method[1].checked)//~x
  {document.edit.former.disabled=1;document.edit.formerx.disabled=1;//~P
  document.edit.formers.disabled=1;document.edit.formersx.disabled=1;}//~Q
}
```

其中，edit 是表单名；mode 是用户名 ridio 控件的名字；method 是 Modify ridio 控件的名字。

(5)function CoverUncover()在程序中的应用[5]：

```
<html>
<head>组合逻辑在编程中的应用</head>
<body>
<SCRIPT LANGUAGE="javascript">
function CoverUncover() //蛰伏或激活 text 控件
{//源码如上，此处省略
}
</script>
<form name=edit><table>
<tr><td><b>Create</b>
  <input name=method type=radio value=0 checked onClick='CoverUncover()'></td>
  <td><b>Modify</b>
  <input name=method type=radio value=1 onClick='CoverUncover()'></td></tr>
<tr><td><b>旧用户名</b></td>
<td>U::from<input name=former size=9 value=bc05142001 disabled=1>
    to<input name=formerx size=9 value=bc05142005 disabled=1>
    OR<input name=formers size=7 value=qb040820 disabled=1>
+<input name=formersx size=7 value=01,02,05 disabled=1></td></tr>
<tr><td><b>用户名</b></td>
<td><input name=mode type=radio value=1 checked onClick='CoverUncover()'>
  from<input name=user size=9 value=vc05142001>
  to<input name=userx size=9 value=vc05142005>
  <input name=mode type=radio value=0 onClick='CoverUncover()'>
  <input name=users size=7 value=vb040820 disabled=1>
  +<input name=usersx size=7 value=01,02,05 disabled=1></td></tr>
</table></form>
</body>
</html>
```

总之，上述步骤如下：

①合理设定逻辑变量和函数；

②简化逻辑函数表达式；

③转化为伪码；

④用程序语言表达。

2 结束语

这种方法具有一般性意义。

①不用为每个 onClick 事件分别编写源代码，仅用一个 CoverUncover()函数。统一编写，控制简捷，源码较少，步骤清晰，易调试和修改。

②对其它的一些显示控制、程序流控制等也有效。

参考文献：

[1] 耿素云，屈婉玲. 离散数学[M]. 北京：北京大学出版社，2002.67-68.

[2] 江晓安. 计算机电子电路技术－数字电子部分[M]. 西安：

(下转第 159 页)

应用时,数据集中的数据来自 OLAP 后得到的结果。为叙述方便,笔者构造了饼图的数据集。表 1 为构造程序。

表 1　数据集构造程序

```
DatasetProducer pieData = new DatasetProducer() {
  public Object produceDataset(Map params) {
  //建造饼图的数据集
  DefaultPieDataset ds = new DefaultPieDataset();
        ds.setValue("一类烟", 231500);
        ds.setValue("二类烟", 256458);
        ds.setValue("三类烟", 145828);
        ds.setValue("四类烟", 578828);
        ds.setValue("五类烟", 142828);
          return ds;
        }
  };
  ……
  //将得到的对象类赋给属性对象,供以后调用时使用
  pageContext.setAttribute("pieData1", pieData);
```

在表 1 中,首先建造了一个 DatasetProducer 接口实现对象 pieData。通过 pieData 类的 produceDataset 方法建立了一个数据集,之后将这个 pieData 对象赋值给系统属性 pieData1 供后面的程序调用。

在显示页面(JSP 页面)中使用如表 2 中所示语句对数据集进行调用。

表 2　图形显示程序

```
<%@page contentType="text/html"%>
<%@taglib uri='/WEB-INF/cewolf.tld' prefix='cewolf' %
>
                ……
  <cewolf:chart id="pieChart" title="Pie" type="pie">
    <cewolf:gradientpaint>
      <cewolf:point x="0" y="0" color="#FFFFFF" />
      <cewolf:pointx="300" y="0" color="#DDDDFF" /
>
    </cewolf:gradientpaint>
    <cewolf:data>
        <cewolf:producer id="pieData1" />
    </cewolf:data>
  </cewolf:chart>
    <cewolf:img chartid="pieChart" renderer="/cewolf" width
="300" height="300"/>
```

由<%@taglib uri='/WEB-INF/cewolf.tld' prefix='cewolf' %>知系统使用了 cewolf.tld 标签库[5]。在 JfreeChart 中,所有的页面显示内容都是通过标签调用来完成的。

在表 2 中,<cewolf:chart>标签定义了图形的数据源及显示格式。其中在<cewolf:gradientpaint>子标签中定义了图形的背景,渲染图形的显示效果;<cewolf:data>标签中定义了数据源。由<cewolf:producer id="pieData1" />可知其使用了前文定义的数据源。

最后,在<cewolf:img>标签中定义了待显示图形 id 值、目标路径、高度、宽度等信息。

3.4　系统显示效果

图 4 为以上程序最终的页面显示效果。

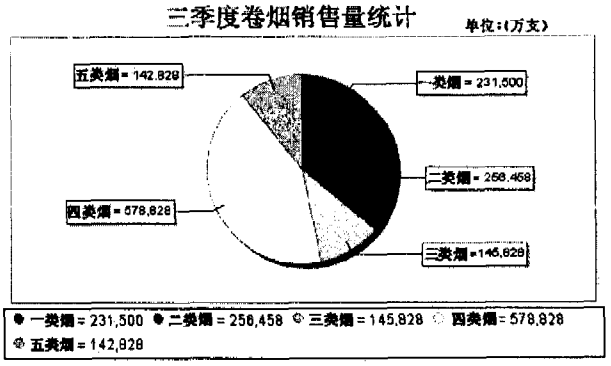

图 4　饼图显示效果图

可以看到,各数据项的值占总值的比例通过其所占饼图的面积的大小形象地表现出来了。在各数据项处用标签标出其名称及其数值。下部是各数据项对应颜色及名称、数值的说明。

除了饼图外,JfreeChart 还可以显示曲线图、折线图、直方图等常用图形。

4　结　论

随着数据库技术的迅速发展及 Web 系统的广泛应用,人们希望以更直观、明确、易于接受的形式获取数据。图形组件技术必将成为 Web 应用不可缺少的一部分。

参考文献:

[1] 冀振燕. UML 系统分析设计与应用案例[M]. 北京:人民邮电出版社,2003.52-94.

[2] 尤克滨. UML 应用建模实践过程[M]. 北京:机械工业出版社,2003.14-34.

[3] 阎　宏. JAVA 与模式[M]. 北京:电子工业出版社,2002.127-136.

[4] Gilbert D. The JFreeChart Developer Guide[Z]. [s.l.]:Object Refinery Limited,2004.

[5] Turner J, Bedell K. Struts kick start[M]. Indiana: Sams Publishing,2002.183-194.

(上接第 156 页)

西安电子科技大学出版社,2004.26-35.

[3] 王诗兵. 关于卡诺图法实现逻辑函数变换的研究[J]. 安徽职业技术学院学报,2005(1):5-7.

[4] 殷人昆. 软件工程复习与考试指导[M]. 北京:高等教育出版社,2001.110-111.

[5] 李　宇. JavaScript 网页特效实例解析[M]. 北京:机械工业出版社,2003.198-206.