

商品标签报表系统分析与实现

the Research and Discussion of the Realization Method of the Report System for Standard Label

高焕兵 袁丽艳 程晓辉 (山东建筑大学信息与电气工程学院)

摘要:随着质量监督系统要求的提高,开发一些对商品标签进行规范化的软件的需要越来越迫切。本文以某标准标签报表系统的项目需求为基础,详细介绍了对这一标准标签报表系统的分析和设计。然后对各个功能的实现提出了几种实现方法,并对这些实现方法作了分析和比较。重点比较了顺序窗口的实现方法,选定了一种应用在了系统实现中,并详细介绍了这种方法的实现思路 and 具体程序。实践证明,这一系统在标准化商品标签的应用中是合理并行之有效的。

关键词:商品标签 报表 模态窗口

1 引言

在生活中经常遇到商品的标签不合格而误导消费者的情况,因此相关单位使用一些标准标签制作系统来规范监督商品标签就迫在眉睫。另一方面,很多生产厂家也经常遇到在向国家质监系统申报商品标签时因为申报样品的不规范而被退回的情况,如果使用标准标签制作系统来制作样品,将会节省大量人力和时间,提高生产效率。本文以某省质监单位的需求项目为背景,介绍了一个以 Delphi 为开发环境的标准标签制作系统的实现方法,以及对其实现方法的一些分析探讨。由于篇幅所限,本文只介绍其食品标签制作系统子模块。

2 系统需求及分析设计

根据国家相关规定,企业设计、制作预包装食品标签必须遵守以下原则要求:

(1) 食品标签的所有内容,不得以错误的、引起误解的或欺骗性的方式描述或介绍食品;

(2) 食品标签所有的内容,不得以直接或间接暗示性的语言、图形、符号,导致消费者将食品或食品的某一性质与另一产品混淆;

(3) 食品标签的所有内容,必须符合国家法律和法规的规定,符合相应产品标准的规定;

(4) 食品标签的所有内容,必须通俗易懂、准确、

科学,不能使用深奥难懂、生造的、谁也无法理解的术语和形容词。^[1]

但据调查,市场上的食品标签存在很多不规范现象,为此我们需要做出一套标准标签的生成系统,使食品生产厂家只需要填写相关表单项,就可以生成标准的食品标签。鉴于不同食品的标签样式并不能做硬性规定,本系统只能生成标准的标签内容,以及对其中的文字和图标字体、大小、颜色等的要求,并不是生成可以直接印刷使用的标签。

基于以上需求,我们只需要开发一个单机版的软件即可,所以我们不需要采用支持分布式架构的开发工具。Delphi 以其高效的开发效率而被广泛使用,所以本系统也决定使用 Delphi 开发。Delphi 目前最高版本为 Delphi 2005,已经是一个很成熟的分布式系统的开发环境,提供了对微软 .net 框架的支持,但是本系统并不需要这些特性,所以可以采用 Delphi 的对 win32 程序支持最好的 Delphi 6.0 或者 7.0 版本^[2,4]。因为本系统要作为单机发布版本,必须使用户的安装简单灵活,所以数据库管理系统采用 Access,访问引擎使用 ODBC,并在系统安装或者启动的时候动态配置 ODBC 数据源。

一个食品标签包含的内容可能有很多,因此我们用一个窗口有可能不能展示所有应填表单,这就需要把各项内容安排在几个窗体上。每个窗体上应该包含

的表单应该先按相近和相似性分类显示,然后再考虑整体界面的安排。分析所有的食品可能出现的表单项,可以把窗体分为以下几个:

(1) 食品基本信息:包括名称,重量或容量,生产日期,保质期,存放方式等。

(2) 食品生产厂家基本信息。包括生产厂家的名称,性质,集团公司,具体子公司或生产基地,地址,邮政编码,电话等。

(3) 成分或者配料信息。

(4) 食品所具有的图标。包括:QS 证号及图标,条形码,注册商标,中国名牌,绿色食品,保健食品等图标。

为了方便用户的使用,我们可以把这些窗体按顺序排列,用户填写完一个窗体的表单之后,点击一个“下一步”按钮可以进入下一个窗口继续填写生于内容,并且可以点击一个“上一步”按钮来返回上一个窗体查看或者修改已经填写的内容。在最后一个窗体填写完毕之后,就可以对标签内容进行预览和打印了。

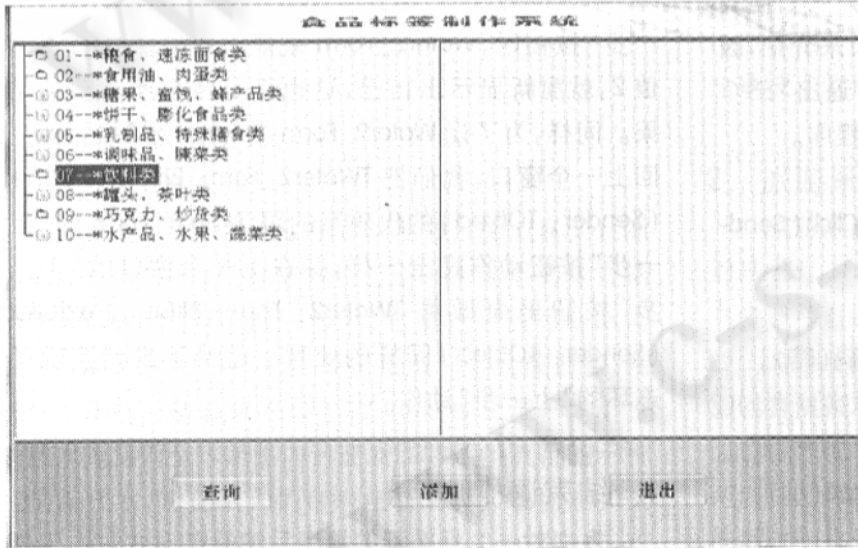


图 1 系统主窗口

3 系统功能实现

本系统的实现主要有三个要点:动态配置 ODBC 数据源、最后的预览技术和各个窗体的顺序显示。ODBC 的数据源的配置和报表预览已经有成熟技术,在此不再赘述。

对于实现“上一步”、“下一步”的顺序窗体,实现

的方法有很多。例如,可以用一个 PageControl 来作为母体,然后把各个窗体的内容分别作为 PageControl 的几个 TabSheet,我们可以把上面的标签隐藏起来,然后在 PageControl 所在的窗体上放置“上一步”、“下一步”两个按钮,分别对应 PageControl1.ActivePageIndex = PageControl1.ActivePageIndex - 1 和 PageControl1.ActivePageIndex = PageControl1.ActivePageIndex + 1 两个动作,就能给用一种在不断切换窗体的感觉。这种方法的优点是实现简单,显示的每个窗体的大小能保证严格的一致;缺点是所有内容全部放到了一个 PageControl 上,是 PageControl 的体积过大,一旦不小心破坏了这个 PageControl,那么将会丢失所有的内容,而且因为各个窗体包含的内容有多有少,全部用同一大小的窗体显得很失调。因此我们应该把各个窗体作为独立的窗体来设计,但是如何实现各个窗体之间的顺序跳转呢?下面我们按照系统的操作流程来介绍其实现方法。

系统启动之后,主界面上应该显示各种食品的分类列表,方便用户直接选择创建新的食品的分类。这种分类显示我们可以采用微软管理控制台(MMC)的样式,即窗口左半部分为树形目录,右半部分为选中项目的具体内容。如图 1 所示。

在主窗口上选择好分类之后,通过点击“添加”按钮来开始添加新的食品标签,首先弹出第一个窗口,即前面讨论的“食品基本信息”窗口。因为这个窗口为第一个窗口,所以它只具有“下一步”和“退出”两个按钮:点击“下一步”保存本页内容并继续定义后面的项目;点击“退出”,结束新标签的定制,本页内容不保存。

主窗口的“添加”按钮对应的事件监听代码如下(只取与本文相关部分):

```

procedure TMain_Form. BitBtn2Click( Sender: TObject );
begin
    Foodid := ;
    if pos( '070201', typeid ) > 0 then // 根据选择的类别决定第一个定义页面的类别

```

```

begin//070201 是对饮料类的编号,所以显示
Drink_Form
    Drink_Form := TDrink_Form.Create( nil );
    Drink_Form.ShowModal;
    Drink_Form.Free;
end
outline1.OnClick( outline1 );
end;

```

代码的关键流程已经做了注释,在这里我们是让“食品基本信息”窗口 ShowModal,即模态显示,即主程序要等待这个窗口关闭才继续执行,并且在关闭时 ShowModal 回返回一个数字,这个数字可以是一个系统默认的数值,但这样缺乏灵活性。程序员可以通过对模态窗口的 ModalResult 属性赋予任意自定义的数值,这个结果就会是 ShowModal 的返回值,这样我们就可以达到灵活分支控制的目的。我们正是利用了 ShowModal 函数的这一特点来实现窗口的顺序或者逆序显示的。填写完“食品基本信息”窗口后,就需要对“下一步”按钮做进一步处理。为了集中介绍如何实现窗口的顺序显示,我们选择一个中间窗口来介绍,因为中间窗口具有“上一步”、“下一步”和“退出”三个按钮。下面先给出这三个按钮的事件监听程序。

“下一步”按钮:

```

procedure TWater2_Form.BitBtn_NextClick( Sender: TObject );
begin
    {……………表单验证、内容存储等代码(略)}
    hide;
    CFLabel := true;
    Water3_Form := TWater3_Form.Create( nil );
    case Water3_Form.ShowModal of
        1: ModalResult := 1;
        2: Show;
    else
        ModalResult := 3;
    end;
    Water3_Form.free;
end;
“上一步”按钮:
procedure TWater2_Form.BitBtn_PrioClick( Sender:

```

```

TObject );
begin
    ModalResult := 2;
end;
“退出”按钮:
procedure TWater2_Form.BitBtn_QuitClick( Sender:
TObject );
begin
    ModalResult := 3;
end;

```

在 BitBtn_NextClick 的处理中,我们首先调用 hide 隐藏本窗口 Water2_Form,然后创建下一个窗口 Water3_Form 并使其按模态方式显示,于是 Water2_Form 就处于等待状态,直到 Water3_Form 被关闭,进而判断 ShowModal 的返回结果:如果等于 1 则把 Water2_Form 窗口的模态显示返回结果设置为 1;如果等于 2 则重新把已经隐藏的 Water2_Form 显示出来。这样只要我们 Water3_Form 的“上一步”按钮动作设置为 ModalResult := 2,用户在 Water3_Form 上点击“上一步”按钮时,Water2_Form 就能够等待到这个返回值 2,就重新显示出自己,这也正是我们所需要的结果。同样,为了让 Water2_Form 的“上一步”按钮能返回上一个窗口,我们把 TWater2_Form.BitBtn_PrioClick(Sender: TObject) 的代码写的跟 Water3_Form 的“上一步”按钮动作完全一样,并在上一个窗口的“下一步”按钮里面做跟 TWater2_Form.BitBtn_NextClick(Sender: TObject) 同样的处理。递推下去就实现了连环的“上一步”动作。

4 结束语

本文对一个标准标签报表系统的实现方法作了分析和比较,综合其利弊,选定了其中一种。希望能在以后的实践研究过程中能得以补充和修正。

参考文献

- 1 GB7718—87,食品标签通用标准[S].
- 2 王晟、吕宗智、王晟,Delphi 实用技术精粹[M],北京人民邮电出版社,2000.
- 3 李维,面向对象开发实践之路——Delphi 版[M],北京电子工业出版社,2005年.