

检测类数据库的软件重用技术研究^①

Research on Software Reuse of Testing Database

李亚民 吴开华 (杭州电子科技大学 自动化学院 浙江 杭州 310018)

摘要: 在重用检测类数据库软件时,所遇到的重要问题之一是 DataWindow 的列标题修改问题。因为目前的应用中,不能实现由普通用户自行完成对它的修改,因此成了此类数据库软件重用过程中的主要障碍之一。本文首先构建了特殊的底层数据库,然后通过数据库的初始化、更新及 DataWindow 的列标题更新,最终实现了 DataWindow 的列标题修改。应用本技术开发检测类数据库软件,解决了不同检测对象下数据库软件的重用性问题,减小了软件的开发时间,方便简单,易于操作。

关键词: 软件重用技术 数据库 DataWindow

1 引言

软件重用技术是指在开发软件系统时,重复使用已有的软件成分,该软件成分可能是已存在的软件,也可能是专门设计可重用的构件^[1]。根据目前检测类数据库软件的制作现状,存在一个数据库构造、主要功能、显示界面、制作过程等可重复的模式,因此应用软件重用技术可以使检测类软件的开发过程大大简化^[2]。但是当检测对象不同时,要求的检测参数名称一般也不同,表现为 DataWindow 中显示列标题的不同,因此如何实现 DataWindow 中的列标题修改就成了主要问题之一。目前,此问题的主要解决方法是在开发平台中把每一个参数名称列的列标题修改或者删除,但这必须由数据库制作人员操作,不能由普通用户直接完成修改。由此带来的软件修改周期将大大增加,软件的重用性也大大降低。本文所用的 PowerBuilder 工作平台,是面向客户/服务器应用领域的前端开发工具。在此平台上应用软件重用技术可设计出高质量的软件系统^[3,4]。本文就是在 PowerBuilder 工作平台上构建了特殊的底层数据库,然后通过数据库面板对其初始化,结合前端窗口对数据库更新及 DataWindow 的列标题更新等操作,最终实现 DataWindow 的列标题修改。本文以作者所制作的一种检测数据库软件为例,讨论实现方法。

2 总体方案设计

底层的数据库结构设计是本方案的依托,参数名称修改是其外在表现,两者相互联系,不可分割。本文的总体设计构架如图 1 所示:

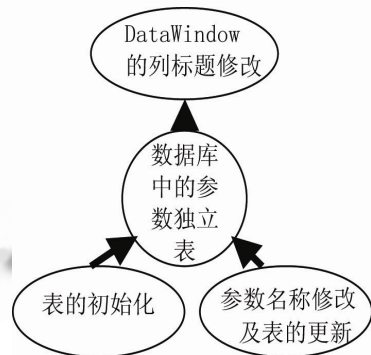


图 1 总体结构图

首先在数据库中构建了参数独立表 (Canshudulib 表),接着对表中的各个字段设计了初始化值,然后通过参数名称修改窗口更新此表中的数据,最后通过刷新或重新打开窗口等操作更新 DataWindow 中的内容。四步操作紧密联系,环环相扣,以下将分别加以描述。

^① 基金项目:国家自然科学基金项目(60870007);浙江省重大科技专项项目(2007C13013)

收稿时间:2008-10-10

3 参数独立表

参数独立表的建立是本方案的核心数据库部分，后面的程序设计都将以此为依托。

建立参数独立表的目的主要有两个：

其一，用于储存各个参数的新名称，以备 DataWindow 的列标题更新之用；

其二，在整个数据库系统中，由于此表储存了各个参数的初始值信息。因此，在其它地方，如实现查询、统计、报表等功能时，就可以使用这些参数的初始值信息，而不用考虑其新名称具体是什么，这样，程序的复杂度、代码长度等都将大大降低。

参数独立表的详细内容如表 1 所示。

表 1 参数独立表

表名	Name	Type	Length	功能说明
Canshudulib	Daima	varchar	20	表的关键字
	Hname	varchar	100	储存原参数的初始值信息
	Htext	varchar	100	储存修改后的新参数名称

4 参数独立表初始化

在 PowerBuilder 的开发环境中，DataWindow 的制作页面如图 2 所示(本图中显示的是 Data Window 的默认设置)。从图中可以看出，列 Canshu2 的 General 选项卡内，Name 项的内容是 canshu2_t，Text 项的内容是 Canshu2。

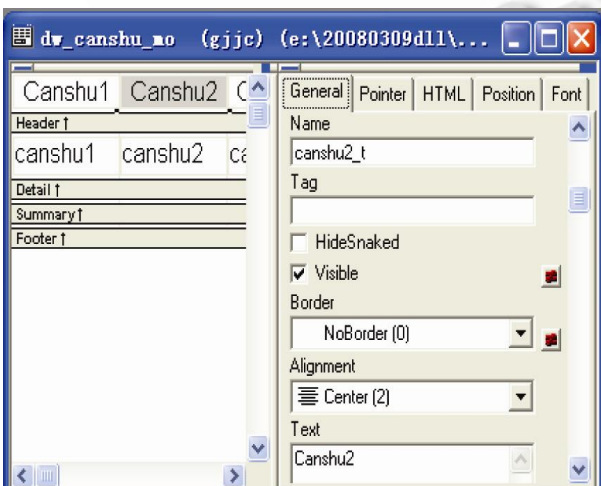


图 2 DataWindow 的制作页面

这里存在一个规律：某列的 Name 项内容为该列列名+_t，Text 项中的内容为列名(首字母大写)，并且两者均为默认值，其中 Text 项的内容就是该列的列标题，也就是最终显示在窗口中的内容。

本文就是根据此规律，在表 Canshudulib 的列 Hname 中存储了此原始联系信息，即：代表原始参数的列名+_t。因此，文中把列 Hname 中的内容初始化为代表原始参数的列名+_t；列 Id 的内容是从数值 1 开始的参数按照顺序逐个递加的数值；列 Htext 中的值可初始化为任意值，可供用户修改的就是此列的值。此表的内容在建立之初就应该按此规律对其初始化，如图 3 所示(本文使用的是 SQL 数据库)。

hname	htext	id
canshu1_t	参数1	1
canshu2_t	参数2	2
canshu3_t	参数3	3

图 3 初始化后的参数独立表(部分行)

5 参数名称修改

参数名称修改是直接为用户相关的显示窗口部分，也是修改数据库数据的入口部分。其工作原理是：读取表 Canshudulib 中的数据到窗口显示，并提供功能按钮以由用户来修改表中某项内容，其流程图如图 4 所示。

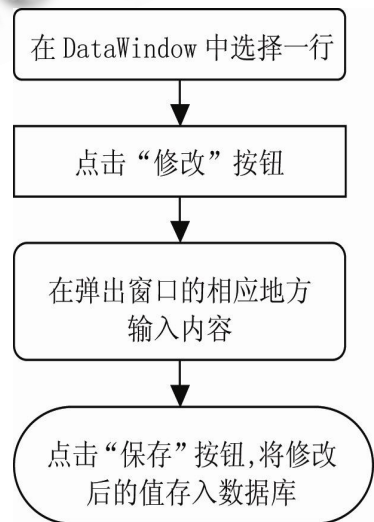


图 4 参数名称修改流程图

以下是参数修改的主要实现代码:

w_canshu_xiugai 窗口的 cb_1 控件(“修改”按钮)的 Clicked 事件中的代码:

```
long ll_row
header_name h_nt //header_name 为自定义结构体
ll_row=dw_1.getrow()
if ll_row=0 then
    messagebox('提示','请选择一行!')
    return
end if
h_nt.text=dw_1.getitemstring(ll_row,"htext")
h_nt.name=dw_1.getitemstring(ll_row,"hname")
openwithparm(w_canshu_popup,h_nt) // 打开 w_canshu_popup 窗口, 并传递参数 h_nt
w_canshu_popup 窗口控件的 Open 事件中的代码:
h_nt=message.powerobjectparm
sle_1.text=h_nt.text
sle_1.enabled=false //不让用户修改 h_nt.text 参数的值
sle_2.setfocus()
w_canshu_popup 窗口的 cb_1 控件(“保存”按钮)的 Clicked 事件中的代码:
if sle_2.text="" then
    messagebox("提示","请输入一个值!")
    return
else
// 更新 canshubiao
UPDATE canshubiao
SET htext = :sle_2.text
where hname=:h_nt.name;
if sqlca.sqlcode<>0 then
    messagebox("提示","名称修改未成功, 请检查数据库连接!")
    rollback using sqlca;
else
    commit using sqlca;
```

```
end if
```

```
end if
```

```
close(parent)
```

6 DataWindow的列标题更新

DataWindow 的列标题更新就是把第三步中最新输入的内容反映到显示窗口中, 其工作原理为: 把表 Canshudulib 中列 Hname 的内容读取出来, 并替换当前 DataWindow 的列标题。此时, 窗口显示的 DataWindow 列标题就是用户修改后的内容。

实现此操作的时机是在软件启动后第一次打开 DataWindow 时, 在该 DataWindow 的 Constructor 事件(作用为构造 DataWindow)中完成。或者使用单独的更新功能按钮完成此功能: 在此按钮的 Clicked 事件中触发该 DataWindow 的 Constructor 事件。

其主要实现过程是: 连接数据库, 得到 Canshudulib 表中的最新内容, 把列 Hname、Htext 值赋予变量, 然后根据所得固定顺序和 Hname 值, 把 Htext 值赋予当前 DataWindow 中相应列的 text 内容中。以下是其主要实现代码。

实现 DataWindow 的列表名更新函数 gf_genxincanshu(datawindow dw_n)的主要代码:

```
//dw_c 是 DataWindow 型参数
int i
int n=100 //此处表明表中有 100 行, 此值可修改
string header_name,header_text
for i=1 to n
把每一行的值选入参数 header_name,header_text 中
SELECT canshubiao.hname,
        canshubiao.htext
INTO :header_name,
        :header_text
FROM canshubiao
WHERE canshubiao.id = :i ;
//根据得到的 header_name 值, 用相应的 header_text 替换 dw_c.object.canshu1_t.text 的值
```

(下转第 185 页)

(上接第 150 页)

```
choose case header_name
    case "canshu1_t"
dw_c.object.canshu1_t.text =header_text
    case "canshu2_t"
dw_c.object.canshu2_t.text =header_text
    .....
end choose
next
```

在所有需要更新列名称的 **DataWindow** 的 **Constructor** 事件中写入以下代码:

```
gf_genxincanshu(dw_1) //调用此函数以实现
列标题更新
```

7 结论

本文通过底层数据库结构上的设计,结合参数名称修改、**DataWindow** 列标题修改等过程,成功地实现了 **DataWindow** 中列标题的修改问题。按照这种方法制作的软件,将可以由无数据库软件开发经验者或

者普通用户根据不同的检测对象直接修改,而不需要专业的数据库软件开发者修改或者重新制作软件,从而减小了数据库软件的开发时间,提高了软件通用性。经实际使用表明,本方法有效地解决了检测类数据库软件的此类重用性问题。本文在重点解决 **DataWindow** 中列标题如何修改基础上,希望对检测类数据库软件的重用技术研究有所帮助。

参考文献

- 1 Jacobson I, Matin G Software reuse: architecture, process and organization for business success.北京:机械工业出版社,2003.
- 2 罗军,江成华.数据库结构复用技术研究.计算机工程与设计,2005,26(11):3010-3012.
- 3 柯建勋,张涛,邵亮,刘建刚.PowerBuilder 8.0 进阶篇.北京:清华大学出版社,2002.
- 4 张显全,蒋联源,唐振军.基于 PowerBuilder 的软件重用研究.软件时空,2006,22(6-3):298-300.