

自动生成特定伪码的设计与实现

周雪莲, 罗代升, 张朋, 张天宇, 王博

(四川大学图像信息研究所 成都 610064)

【摘要】针对屏幕显示设计效率较低的情况,完成了自动生成用户交互界面伪码的设计,实现了自动生成屏幕显示伪码的系统。该系统可以让用户方便地设计出任何类型的屏幕显示界面。应用结果表明该系统可以缩短软件项目的开发周期,提高设计者的工作效率。

关键词 数据库; 模块; 屏幕显示
中图分类号 TP311 文献标识码 A

Design and Realization of Automatically Creating Particular Fake Code

ZHOU Xue-lian, LUO Dai-sheng, ZHANG Peng, ZHANG Tian-yu, WANG Bo
(Image Information Institute, Sichuan University Chengdu 610064)

Abstract An automatic On Screen Display(OSD) fake code generation system that addressed the problem of ineffective OSD design is proposed and realized in this paper. The system could be used to design any type of OSD interface in a hassle free way. The application result indicates that software development period could be shortened by this system efficiently.

Key words database; module; on screen display

1 屏幕显示技术

1.1 背景

运行在机顶盒上的软件大都是基于C语言的, C语言是面向过程化的语言,其代码量(尤其是其屏幕显示(On Screen Display, OSD)部分的设计^[1])相当大。OSD是指将所有图像控制效果直观地显示在屏幕上,为用户提供简单而全面的调控能力^[2]。用VC和Java实现的窗口、面板等控件,在C语言里需要很长的代码来实现,程序结构纷繁复杂,费时费力,因此需要一种方法来改进OSD设计。

文献[3]完成了基于C语言的OSD设计的伪代码规范,提供了一个driver。该编译器能识别并支持该driver表达出的各种UI规范。只要有足够的伪代码文件,并将这些伪代码文件作为输入传给该driver,程序就能自动按照该规范分析执行这些伪代码,并画出相应的图像,实现OSD的开发。

本文通过研究规范中的每个模块,生成足够的伪代码文件。设计目标是:输入设计者的绘图,输出一个有效服从规范的目标文件(*.txt文件)。

1.2 相关工作

OSD的实现是先用基本C语言写出基本的绘图函数,再用该绘图函数按先后顺序过程化地绘出各个图形,这些图形组合起来就构成OSD界面。各个界面的切换都按照模块函数的先后调用,如清除、画图形,所有这些过程都要程序员编写代码实现。

已有的成果是这个伪代码的规范,还有VC++6.0开发工具和一些图片资料。该规范定义了各个图形对象的类型、名称、操作码、格式以及相应的参数。下面是规范的部分片断:

```
$BEGIN_INTERFACES$
$BEGIN_DEFINES$
DEFINE_FONT opcode(001) idx(1|...|10) "face"
size hex_atts
DEFINE_PALETTE opcode(002) rrgbb1 ...
rrgbb32
$END_DEFINES$
LOAD_IMAGE opcode(010) idx url ...
$END_LOADING$
$BEGIN_BASIC_UI$
```

```

DRAW_LINE opcode(020) x y w h idx_bgcolor
DRAW_RECT opcode(021) x y w h idx_bgcolor
DRAW_FILLRECT opcode(022) x y w h idx_
bgcolor
DRAW_CLEARRECT opcode(023) x y w h ...
$SEND_BASIC_UI$ ...
$SEND_INTERFACES$

```

本文对规范中的每个模块展开研究,生成足够的伪代码文件。实现方法如图1所示。

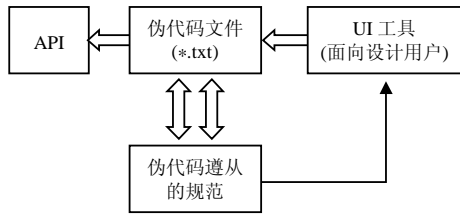


图1 系统实现结构图

2 系统设计

2.1 概要设计

本文采用MFC多文档模板^[4],系统程序流程图如图2所示。

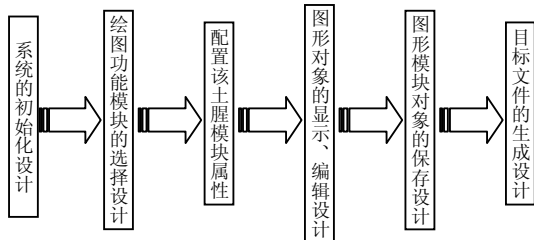


图2 系统程序流程图

2.2 详细设计

1) 数据库的设计:采用MicroSoft的Access创建一个数据库模板^[5],名字为UiData.mdb,存放路径为\TemplateDB\UiData.mdb。创建DEFINE_TABLE、LOADING_TABLE、BASIC_UI_TABLE、FUNCTIONALITY_TABLE和METAINFO_TABLE等五个表,分别对应规范中的五个部分;资源主要定义所需的各种资源(如位图资源、字体等);资源加载部分主要在程序启动时加载各种所需的资源;基本方法部分,主要定义设计UI所用的足够的方法(函数),包括方法名、必须的参数、语法等;其他特殊功能模块部分,该类功能通常用得少,又较复杂,所以需要特殊定义;信息模块部分,用于在设计时跟踪Bug产生的信息。

2) 系统的初始化设计:设计过程主要包括两种情况。(1) 首先创建一个新项目文档,在用户对文档操作前,设计一个对话框供用户输入项目名称。用

户输入的名称既是该项目及其对应数据库的名称,也是输出目标文件(*.txt)的名称。然后,从模板数据库目录中拷贝一个数据库文件到工作数据库目录.\MyProjects,并命名为对应项目名称。如,用户输入项目名称为menuProject,则该项目所对应的数据库为menuProject.mdb,生成目标文件menuProject.txt后保存至目标目录.\MyResult里,然后连接到该数据库,并打开各个表。(2) 打开项目文档,用户通过文件对话框打开一个Project时,系统根据项目文件中读出的信息自动确定该项目对应的数据库路径和名字,并连接到该数据库文件,打开各个数据表,同时自动确定对应的目标文件。

3) 绘图功能模块的选择设计:首先设计一个类作为所有模块类的父类,其原则是一个模块对应一个子类,各个子类有其特殊的参数和方法。根据要求,需要设计的模块有:Line、Rectangle、RoundRectangle、Ellipse、Graph、Button List、Menu List、Edit List、ViewList、Process Bar、Scroll Bar等,这些模块分别对应一个类。为了能编辑修改某个对象,分别为各个模块另外设计一个属性框类,每个模块就包括一个绘图类和一个属性框类。

各模块类的主要成员变量有:坐标参数、宽高参数、使用的位图路径和对象类型。其方法有:绘画方法、选中方法、改变大小方法、拖动方法和弹出属性框方法。由于应用的模块化很强,所以程序的模块化也要很强,即能方便地添加或删除某功能模块,而不影响其他模块。当用户选中某个图形模块时,程序作好绘画该模块的准备,当用户决定动作时,就创建一个该类的对象,并将该对象添加到序列化中。

4) 配置该图形模块属性:每个模块对应的两个类分别是绘图类和属性框类。属性框的弹出由一个菜单项触发,即用户选中某个对象后,单击右键,弹出菜单,触发菜单项“属性”后,就触发了对象的属性框。这个方法是从父类继承过来的虚拟的方法,各个子类继承该方法,创建自己的属性对话框。属性对话框的初始化数据由该对象所属的绘图类的成员参数提供,即绘图类和属性对话框类存在着交互,它们是“近亲”关系。属性框以确认方式关闭时,将更新后的数据反馈给绘图类。

5) 图形对象的显示、编辑设计:图形模块对应的类除了绘图类和属性对话框类之外,还为其设计了一个CobArray类的对象。这个对象动态地保存了该图形模块类所有对象的属性和数据。显示时,遍

历所有CobArray对象,使每个对象具体到每个图形模块对象,再调用其绘图函数,从而将各个图像对象显示出来。判断、编辑某个对象的方法同上述显示的方法类似。

6) 图形模块对象的保存设计:将各图形模块序列化到项目文件里,供再次打开。序列化的内容和方法要具体到每个子类,再按先后顺序动态地将各图形对象的必要属性数据写入相应的数据库表里。

7) 目标文件的生成设计:根据前面用户已确定好的文件名,以写/无创建的方式打开该文件,然后按相应顺序读取数据库各个表的内容。根据操作码整理好要写入目标文件的每行命令的字符串后,将这些字符串逐行写入目标文件,最后通过序列化保存项目文档。

3 系统实现

3.1 添加各个模块类

设计中的模块类包括:直线类、矩形类、圆角矩形类、椭圆类、画位图类、按钮列表类、菜单列表类、可编辑列表类、视图列表类、进度条类、滚动条类等,其中直线类、矩形类、椭圆类、位图类共有个属性框类;其余各个类分别对应一个独立的属性框类。模块类及对应的属性框类实现该对象的属性设置、属性保存、显示方法和对象属性伪码的生成等功能。当用户选中某个特定图形类别后,系统能够识别到该图形所属的类,然后创建该类的对象,再根据用户的需求与交互,该对象调用自己特有的方法进行图形的绘制、修改及删除等,同时记录下该对象的各个参数。

3.2 写数据库

数据库采用Microsoft Access数据库,包括数据库文件的设计(在Windows上进行,设计时注意各列项的数据类型、关键字、可否重复,尽量避免中文注释),采用VC专有的读写数据库接口。用户要保存项目时,将所有已生成对象的必要参数写入数据库,为后面写目标文件做准备,根据数据库的内容分析、整理,并用规定的格式将数据写入目标文件。

3.3 写目标文件接口

写目标文件包括向目标文件写入用户定义的资源及格式等内容,其主要接口函数如下:

WritePseudocodeIntoFile 写伪码至目标文件(父函数)

Write_FormatCmdLine 写格式行到目标文件

WriteDefinePartIntoTargetFile 写定义部分至目标文件

Write_DefineFont_CmdLine_TargetFile 写定义之字体部分至目标文件

Write_DefinePalette_CmdLine_TargetFile 写定义之调色板至目标文件

Write_DefineFeature_CmdLine_TargetFile 写定义之特征至目标文件

WriteLoadPartIntoTargetFile 写加载资源部分至目标文件

Write_LoadImage_CmdLine_TargetFile 写加载位图部分至目标文件

WriteBasicUiPartIntoTargetFile 写基本图形对象至目标文件 ...

3.4 生成伪代码文件

生成足够的伪代码文件后,将这些伪代码文件放在一个特定的目录下,以文件名为参数,运行已有的底层driver批处理程序,生成C语言能识别的目标代码,从而转化为符合用户设计的OSD界面,大大缩短了OSD的设计时间,提高工作效率。

4 系统运行结果及分析

根据上述设计实现伪码生成系统,其主设计平台如图3所示。运行后生成的伪码文件如图4所示。



图3 系统主设计平台

```
SECTION_INTERFACE
SECTION_DEFINES
SECTION_PALETTE
SECTION_LOADIMAGE
SECTION_LOADFONT
SECTION_LOADFEATURE
SECTION_LOADPART
SECTION_DRAW
SECTION_STRING
```

图4 生成的伪代码文件

(下转第324页)

可信度为：

$$C_{CF}(V) = \sum_{i=1}^6 (W_i C_{CF}(P_i)) \quad (7)$$

3 算例

某个精密机械制造企业根据其获得的市场机遇的需要和自己的实际情况，决定联合其他企业共同进行一种专用精密机械产品的设计开发，盟主企业需要对参选企业进行合作可信度分析。经过前期工作，获取某参选企业的 T 层指标可信度矩阵和 T 层指标权重矩阵如下：

$$C_{CF}(T)_{6 \times 5} = \begin{bmatrix} 0.8 & 0.7 & 1.0 & 0.4 & 1 \\ 0.7 & 0.8 & 1.0 & 0.7 & 1 \\ 1.0 & 0.7 & 0.9 & 0.4 & 0 \\ 0.5 & 0.8 & 1.0 & 0 & 0 \\ 1.0 & 0.8 & 0.4 & 0.9 & 0 \\ 0.9 & 1.0 & 0 & 0 & 0 \end{bmatrix}$$

$$w_{6 \times 5} = \begin{bmatrix} 0.137 & 0.253 & 0.300 & 0.168 & 0.142 \\ 0.351 & 0.147 & 0.152 & 0.175 & 0.175 \\ 0.254 & 0.190 & 0.380 & 0.176 & 0 \\ 0.167 & 0.333 & 0.500 & 0 & 0 \\ 0.200 & 0.300 & 0.269 & 0.231 & 0 \\ 0.500 & 0.500 & 0 & 0 & 0 \end{bmatrix}$$

而 P 层指标的权重集合为 $\{0.167, 0.333, 0.250, 0.083, 0.083, 0.083\}$ ， $C_{CF}(P, T)=1$ ， $C_{CF}(V, P)=1$ ，则按照上述方法，依次计算如下：

$$\{C_{CF}(P_i)\} = C_{CF}(P, T) \left[\sum_{j=1}^5 (w_{ij} \times C_{CF}(T_{ij})) \right] = \{0.795 \ 9, 0.812 \ 8, 0.799 \ 4, 0.849 \ 9, 0.755 \ 5, 0.950 \ 0\};$$

(上接第262页)

该系统经运行试验，可生成足够的伪代码文件。将这些伪代码文件放在一个特定的目录下，输给已有的driver后，程序按照规范分析并执行这些伪代码文件，画出相应的图像界面，从而实现OSD的开发。

5 结束语

本文设计的系统可以直接从系统主设计平台设计OSD界面，然后将设计好的界面转化生成相应的伪码，并将伪码植入代码中，让机器去分析这些伪码，然后作出相应的动作，画出图像。该系统缩短了软件项目的开发周期，提高设计者的工作效率，可以使工程师有更多的时间去研发软件底层驱动与

$$C_{CF}(V) = 0.749 \ 3。$$

在实际应用中，常常有多个企业参与竞争。因此，在对合作伙伴进行选择评估时，要计算多家企业的根指标可信度 $C_{CF}(V)$ ，取 $C_{CF}(V)$ 值最大的企业作为合作伙伴。

4 结论

将基于可信度的不确定推理方法运用到动态联盟的伙伴评价中，根据量化指标进行建盟决策，能够客观、全面、快捷地对参选企业进行评价。随着经济全球化和企业竞争的日益加剧，影响伙伴评价的新的约束将不断产生，所以该模型结构和评价方法还应随之不断完善。

参考文献

- [1] ZHUANG Wan-yu, LING Dan, ZHAO Jin. The research of domestic enterprise ability index and appraisalment model[C]//International Conferer on Intelliget Mechatronics and Automation. [地址不祥]: IEEE, 2004.
- [2] 成红, 银路, 周岚. 高新技术行业动态联盟探析[J]. 电子科技大学学报(社科版), 2003, 5(1):12-16.
- [3] 郑文军, 张旭梅, 刘飞, 等. 虚拟企业合作伙伴评价体系及优化决策[J]. 计算机集成制造系统-CIMS, 2000, 6(5): 63-67.
- [4] 林晓通, 王宁生. 企业动态联盟中盟友的模糊综合评价[J]. 机械设计与制造工程, 2002, 1(3): 81-83.
- [5] 许学斌, 顾剑飞, 张新曼. 基于模块化神经网络的动态联盟伙伴优化选择[J]. 微电子学与计算机, 2004, 21(1): 90-92.
- [6] 王永庆. 人工智能原理与方法[M]. 西安: 西安交通大学出版社, 1999.

编辑 孙晓丹

中间件及API部分的设计，具有很大的适用价值。

参考文献

- [1] 甘亮, 余松煜. 数字电视监控系统中的OSD实现[J]. 计算机仿真, 2000, (6): 28-30.
- [2] 马延珂. 基于数字机顶盒的OSD设计[J]. 北京广播学院学报(自然科学版), 2003, 10(4): 52-57.
- [3] 李留英, 王戟. UML statecharts的测试用例生成方法[J]. 计算机研究与发展, 2001, 38: 691-697.
- [4] 刘连喜, 徐惠民. MFC框架中的设计模式分析[J]. 计算机应用与软件, 2005, 9: 50-52.
- [5] 惠燕, 潘煜, 徐光辉. 数据库设计范式[J]. 现代电子技术, 2003, 24: 107-108.

编辑 熊思亮