第10卷 第1期 华东交通大学学报 Vol. 10 No. 1
1993年 3月 Journal of East China Jiaotong University Mar. 1993

CAD系统中的菜单设计

邵平平 李正凡

(计算中心)

摘 要

菜单设计是CAD系统界面的主要部分，是一个系统成功与否的关键。方便灵活的菜单会给用户带来熟悉友好的环境，因而能使系统尽可能地发挥其功能。菜单设计一般包括分层设计和菜单编辑两大部分。本文提出了一种菜单设计的方法，它的每一步骤均体现了交互设计，面向用户的设计指导思想。本方法已在微机上用Turbo C实现，结果表明本菜单设计方法简单明了、方便实用，对于CAD系统设计和用户应用都起到了良好的作用。

关键词：直接；关系；菜单设计；分层设计；菜单编辑；开关号；序组

0 引 言

菜单是CAD系统界面的重要组成部分。菜单设计的好坏在很大程度上影响着系统的成功与否。从系统编程的角度看，系统通过菜单将其功能展示给用户，菜单又是用户理解运行系统的媒介，它应为用户提供灵活、熟悉和友好的环境。因此，菜单设的中心任务是展示系统的功能和提供友好的环境。

CAD系统是一个庞大的工程，其中有许多不同类型的功能。如果采取列表式地直接展示来设计菜单，会给用户带来麻烦，不是最佳的组织方法，大系统很少采取这个方法。利用CAD系统功能本身的类型，将同类组合在一起作为显示单元，不同的类设置在不同的层上。这种分层设计既有效地组织了功能序列，又能便于用户理解和记忆这些功能，为用户提供了较好的系统运行环境。

CAD系统运行环境是指功能的排列和功能名的命名。固定的系统运行环境不能够满足用户多方面的要求，因此还必须在菜单设计中考虑易于灵活修改的系统运行环境接口。一个良好的设计可以在一定程度上满足系统的更新，延长CAD系统使用寿命。分层设计研究怎样展示系统的功能，而菜单编辑则研究如何满足用户的要求。菜单设计就是分层设计和菜单编辑。

本文于1992年5月13日收到

1 直接菜单设计

在编制CAD系统软件时，就菜单设计考虑在内，这种方法称之为直接法。由此而形成的设计风格称为直接菜单设计。在编写系统软件时利用开关语句嵌套来进行分层设计，而菜单编辑只能靠修改源程序来获得。例如要设计图1所示菜单，编写一段程序如下：

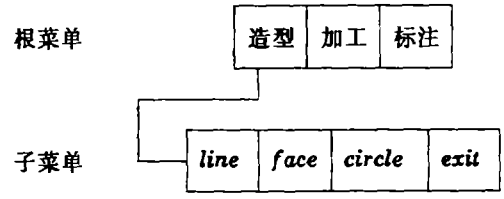

图1　一菜单示意图

```
main ()
{ id=0;
  displaymenu (根);
  while ( id!=-1)
    { id=selectmenu ();
      switch (id)
       { case 造型 : displaymenu (子);
         subid=selectmenu ();
         while (subid!=-1)
               { switch (subid)
                 {case line : line (); break;
                  case face : face (); break;
                  case circle : circle (); break;
                 }
               } break;
         case 加工 : machine (); break;
         case 标注 : dimension (); break;
       }
    }
}
```

其中：machine ()，dimension () 两个函数都可以类似地含有自己的菜单。这是一个典型的直接菜单设计。这种方法的优点是简单明了，缺点是增加和修改菜单都需要在程序上作改动，无法在执行块生成后完成。并且，若某个功能调用需要出现在不同层时，必须在相应的层上分别写入这个功能函数调用。这样会增加程序驻留的内存，影响系统的运行效率。

2 关系菜单设计

分析图1所示菜单，如显示名为face的一项，它处在第二层中，且为第二层的第二个元素，它所调用的函数为face()；完成这个函数调用是通过开关语句的选择。定义开关语句的选择号为此功能的开关号，并把显示名、层次、序号统称为菜单的属性。序组（显示名、层次、序

号）称为菜单项，所有开关号的集合定义为S，所有菜单项的集合定义为M，这样菜单便就是M到S的关系R，用下列等式表示：

$$Menu = R(M,S) \qquad (I)$$

另外，考虑到图1“造型”这一元素项，它仅仅是一个层名，并没有具体的开关号，只对应一个层次数(或层号)。若将层号看作虚开关号，并将所有层号与所有开关号的集合记为SL，则菜单等式I修正为：

$$Menu = R(M,SL) \qquad (I)$$

并非所有M中的元素都是有效的，例如（func，1，1）与（func，1，0）虽是两个不同元素，但它们的显示名是一样的，因而不能同时出现在同一菜单中。

设Mr是某个感兴趣的菜单项集合，此时菜单就是R（Mr，SL），所以菜单是一个对应关系。按这种对应关系来设计菜单就叫做关系菜单设计。确定Mr集合与SL集合，寻找它们的对应关系，这些合起来称层次设计。从一个关系R_1过渡到另一个关系R_2，变动集合Mr与SL，这些合起来称菜单编辑。关系菜单设计灵活、简洁、方便，在不增加SL的情况下，其设计、修改都不用改变源程序。假如在不同的层，例如在l_1和l_2上，都需要同一功能，设其开关号为S，则只要建立对应，

$$(nameofs,l_1,o_1) \to S \text{ 同时 } (nameofs,l_2,o_2) \to S$$

这样便在层l_1第o_1个位置上，在层l_2第o_2个位置上设置名为nameofs的功能调用S。若要在某层，如l_1，设置层l_2的入口，层l_2对应的虚开关号仍为l_2，则建立对应，

$$(nameoflayer,l_1,o_1) \to l_2$$

l_1与l_2可以相同。在l_1层o_1位置上就有一个层名为nameoflayer的菜单。

关系菜单的实现较方便，其程序结构如下：

```
main ()
{ int id, currentlayer;
  createrelation ();
  displaymenu (0);
  id=0;
  while (id!=-1)
     { id=selectmenu ();
       if (id==虚开关号)
          { clearmenu (); currentlayer=虚开关号对应的层;
               displaymenu (currentlayer);}
       else exec-func (id);
     }
}
```

其中，createrelation () 建立菜单关系，displaymenu (id) 显示层号为id的层，即在菜单项(name，l，o) 中，按o的大小次序显示所有l为id的显示名name。selectmenu () 选择当前层中的菜单项，并按对应关系返回此菜单项对应的开关号，clearmenu () 抹去屏幕上的菜单。

exec－func（id）执行 id 所对应的功能调用。

从以上分析可知，菜单关系建立与程序运行是分开的，这样可以设想将菜单关系留给用户建立。系统运行时就读入这个对应关系，其菜单即为这个对应关系。这就是关系菜单设计的精华。

3　关系菜单设计的实现

直接菜单设计的层次设计过程中是同系统一起被实现的，而关系菜单设计并非如此。只是在系统中实现了关系菜单的结构框架，具体的对应关系则留给用户设计。下面几节给出了实现关系菜单设计的一种方法。

3.1　层次设计

开关号集合 SL 被定义为整数集合，其关键是怎样区别实开关号与虚开关号。为此将整数分段，区间〔0，200）分配给实开关号，区间〔200，256）分配给虚开关号，层次为开关号减去200。为了减少内存开销，开关号用无符号变量表示。功能的最大容量为200，层次的最大容量为56。

Mr 集的范围较大，确定它较为复杂。为了使每个不同的功能有不同的显示名，首先在开关号集与显示名之间建立一一对应关系 fname，即开关号 s 的显示名为 fname（s）。然后再建立层次 l 和序号 o 与开关号的对应关系 flayer，即 flayer（l，o）表示（l，o）所对应的开关号，菜单关系为：

$$(fname(flayer(l,o)),l,o) \rightarrow flayer(l,o)$$

这种对应关系可用表1展示。

表1　对应关系表

n_1	name1，name2，……，name n_1				
n_2	MAINMENU	m_1	m_2	…	m_{n2}
n_3	SUBMENU1	S_1	S_2	…	s_{n3}
n_i	SUBMENUi	r_1	r_2	…	r_{ni}
n_1	SUBMENU1	t_1	t_2	…	t_{n1}

表2　图1所示菜单的对应关系表

3	line　face　circle				
4	MAINMENU	201	202	203	204
3	MODELLINE	0	1	2	
0	MACHINE				
0	DIMENSION				

其中，n_1表示实开关号的个数，*name*1对应实开关号0，*name*2对应实开关号1，……，*name* n_1对应实开关号n_1-1，即$fname(s-1)=names$；n_2表示第0层开关号总数，*MAINMENU*是此层的层名，$m_1, m_2, \cdots\cdots m_{n2}$是开关号集合$SL$的一子集，$m_j(j=1,\cdots,n_2)$对应序号$j-1$，用对应关系$flayer(0, j-1)=m_j$表示；$n_i(i=2,\cdots l)$表示第i－2层的开关号个数，Submenui是此层的层名，对应关系为flayer（i－2，j－1）＝r_j。

图1所示菜单可由表2的对应关系表示。

3.2 菜单编辑

这里以表2为基础，说明菜单编辑的步骤。

增加显示名为pocket的功能，则将表2的第一行改为：

4 line face circle pocket

功能的开关号被赋值为3。将此功能设计在MACHINE这一层，则修改第四行为：

1 MACHINE 3

增加清屏功能，它的显示名为clear，为了方便，开关号定为4，并将它设置在每一层上，则整个表2改为表3

表3

5	line face circle pocket clear					
5	MAINMENU	201	202	4	203	204
4	MODELLINE	0	1	2	4	
2	MACHINE	3	4			
1	DIMENSION	4				

增加一层SCREEN，它有功能window、zoom和clear，并且其它层都将这层作为子层，则修改对应关系表为表4。

表4

7	line face circle pocket clear window zoom					
5	MAINMENU	201	202	205	204	203
5	MODELLINE	0	1	2	4	205
3	MACHINE	3	205	4		
2	DIMENSION	4	205			
3	SCREEN	5	6	4		

若要修改显示名，如将clear汉化为“清屏”，circle汉化为“画圆”，则修改表4中的第一行为：

7 line face 画圆 pocket 清屏 window zoom

同样要修改层名，如将MAINMENU改为“主菜单”，则修改第二行为：

5 主菜单 201 202 205 204 203

做上述修改后，原来显示circle、clear的每一个地方都被“画圆”与“清屏”对应替代；出现MAINMENU的地方被改为“主菜单”。由以上分析可得菜单编辑有以下基础命令：

- ·修改显示名 (modify－name)
- ·修改层名 (modify－layername)
- ·修改层号 (modify－layer)
- ·修改序号 (modify－order)
- ·增加一菜单项 (addmenu)
- ·删除一菜单项 (deletemenu)
- ·增加一层 (addlayer)
- ·删除一层 (deletelayer)

3.3 程序实现

为了实现上述菜单设计，设置下列全局变量：

显示名变量：　char *ComName;

层名数组：　char *LayName;

层中菜单项个数：char *LayMaxNum;

层中开关号数组：char *IdCom〔56〕;

功能个数：　char *MaxCom

为了方便用户，将对应关系表作为一文件。它可以被用户修改。readmenu（）读这个文件并建立相应对应关系的数据结构。selectmenu（）用来选择当前菜单项并返回它的序号，displaymenu（i）显示第i层的菜单显示名，clearmenu（）清除菜单。程序编制如下：

```
main ()
 { int currentlayer, frontlayer, order, idswitch;
   currentlayer=0; frontlayer=0; idswitch=0;
   readmenu ();
   displaymenu (0);
   while (idswitch!=-1)
     {order=selectmenu ();
      if (order>=LayMaxNum〔currentlayer〕)
        { if (order==LayMaxNum〔curentlayer〕+1)
                { currentlayer=frontlayer;
                  frontlayer=0;
                  clearmenu ();
                  displaymenu (currentlayer);
                }
          if (order ==LayMaxNum〔curentlayer〕+2)
```

```
                    { currentlayer=0;
                  frontlayer=0;
                  clearmenu ();
                  displaymenu (currentlayer);
                }
              }
          else
              { idswitch=IdCom [currentlayer] [order];
                if (idswitch>=200)
                  { clearmenu ();
                    frontlayer=idswitch-200;
                    displaymenu (currentlayer);
                  }
                else exec-func (idswitch);
              }
      }
    writemenu ();
  }
```

其中，writemenu () 是写对应关系文件，在运行过程中用户能修改菜单，因此必须保存这种修改；exec－func (idswitch) 是一开关选择执行功能函数：

```
exec-func (idswitch)
int idswitch;
{
        switch (id)
            {
               case C: func0 (); break;
               case 1: func1; break;
                 .
                 .
            }
}
```

整个程序用 Turbo C 编写，并在微机上调试通过。

4 结论

菜单是展示 CAD 系统功能和提供 CAD 系统运行环境的用户界面，通过分析，本文将菜单定义为扩展开关号和菜单项之间的对应关系。不同的对应关系给出不同的菜单。扩展开关号被定义为功能的实开关号和层次的虚开关号。菜单项被定义为一序组（显示名，层号，序号）。

关系菜单设计法就是将对应关系程序化，本文给出了一种实现方法。此方法最大限度地为用户提供方便。利用关系菜单设计，摆脱了直接法将菜单看作一颗树的局限，在这里菜单被看作是一有向图。本文提出的关系菜单设计虽可以方便地实现一系列想法，但还是有局限的。未来的研究是要建立菜单项与开关号序组的关系。研究怎样用最小的改动或不改变来增加系统的功能。

参 考 文 献

〔1〕 中国科学院希望高级电脑技术公司. 用C语言开发图形软件. 北京：1991

〔2〕 中国科学院希望高级电脑技术公司. Turbo C 参考手册. 北京：1990

〔3〕 杨光宇编译. Turbo C 高级编程指南. 中国科学院希望高级电脑技术公司，1989

〔4〕 中国科学院希望高级电脑技术公司. Auto CAD 使用手册. 北京：1990

〔5〕 刘应尘译. micro CADDS GCD 3. 0用户手册. 航天CAD技术联合公司，1989

The Menu Designed for CAD System

Shao Pingping　Li Zhengfan

ABSTRACT

Menu designed is major part of CAD system interface. It's one of cruxes of successful system. Convenient and flexible menu can bring users a familiar and friendship interface, so that the system may be made the most of its works. In general, menu designed contains two parts, layer designed and menu edited. In this paper a method of menu designed is advanced. Mutual design and guiding thinking for users are took in its every steps. In micro computer this method is programed in Turbo C. As a result this method of menu designed is concise, convenient and functional, and has fine effect on both CAD system designed and users application.

Key words: direct menu designed; relation menu designed; layer designed; menu edited; switch number; order array