一种动态开放式对话类的设计方法

胡昔祥 郑宁 任域 (杭州电子工业学院 CAD 所 310037)

摘要:本文提出了一种动态开放式对话类的设计方法,它实现了程序与数据的分离,提高了软件的重用性和效率,降低了代码的复杂度。并且利用该技术开发的应用程序具有高度的开放性,只需修改外部数据就可以达到修改数据窗口界面以满足不同要求,从而提高了软件的灵活性和可维护性。它能适用于各种大型复杂软件对对话类窗口的需求,特别是在电信管理网(TMN)的网管系统(NMS)中具有重要的应用价值。

关键词:动态对话类 动态视图 TMN NMS MFC

1. 引言

在 WINDOWS 应用程序中,对话类是最主要的人机界面,他可以提供用户输入信息以及显示机器响应信息。因此,对话窗口是 WINDOWS 程序设计中应用最广泛的对象。特别在大型软件开发中,一般用到大量的相似而又不完全一样的对话窗口,这样就必须创建大量相似的对话窗口对象,做大量不必要的重复工作;而且,这些窗口也不利于管理和维护。通常,各种开发工具都没有提供一种现成有效的开放式动态窗口类。在软件开发过程中,为了提高软件开发效率和软件质量,作者在微软基本类库(MFC)所提供的对话类基础上提出了一种动态开放式对话类。它的设计思想是将应用中的每个窗口对象所拥有的各种组成对象的属性分解归类并存储到数据表中,在需要时,根据具体参数搜索并得到组成窗口的各对象及属性,根据一定的算法和策略动态地生成对话类窗口。该技术在作者从事的大型通信网管系统开发中得到了充分的应用,并取得了良好的效果。

研究动态对话类的目的正是为软件开发人员提供一个开放式的通用对话窗口类,使它具备通用的窗口自动生成和处理功能,并能动态地更新窗口。经过将窗口处理程序与窗口数据分开,不仅提高了代码的重用性和代码效率,而且使窗口更灵活,修改更方便,甚至只要将窗口数据添入数据表中,对话类对象就能自动生成对应的窗口。简化了大量界面处理代码,使得软件人员集中精力于核心处理程序。

2. MFC 对话类的基本原理

MFC(Microsoft Foundation Class)提供了一个基本对话类 CDialog,它是由 CWnd 类派生出来的;因此,它继承了 CWnd 类的大部分属性,即提供了一个窗口所需的基本功能。

在 Visual C++ 中，一个对话窗口对象通常由一个对话模板和一个 CDialog 派生类对象组成。对话模板主要负责窗口控件的显示和布局，对话类对象负责接收和处理来自交互控件和 WINDOWS 系统的消息。当用户与对话类控件交互作用时，控件会发出通知消息给对话类对象，对话类将通知消息交给相应的处理过程。在 CDialog 派生类中可以通过重载来定制自己的消息处理函数；因而 CDialog 派生类具有一定的开放性。

在 CDialog 派生类中每个对话控件对象都有一个成员变量，用于保存用户输入对应控件对象的数据或向用户显示的数据。对话控件对象与它对应的对话类成员变量之间是通过 DDX(对话数据交换)相互作用的，在对话类的构造函数中必须初始化所有的控件对象对应的成员变量，在对话窗口显示之前框架的 DDX 机制将成员变量的值传送给对话控件。同样，当用户点中 OK 按钮，DDX 将对话框控件的值传送给成员变量。如图 1：

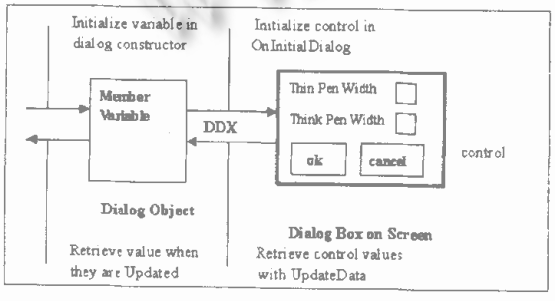

图 1

3. 动态对话类窗口生成算法

动态对话类窗口的组成是不断变化的，即根据当前状态动态地确定当前窗口内容。算法如下：

算法 1：输入：F, A, K, P, R; /* P 为附加属性和条件，R 为窗口规则集 */

输出：D　　/* 组成窗口的数据字段向量 */

DynGenerateWnd(F, A, K, P, R, D)

(1)fi = Select(F); /* 选择功能菜单项 */

Ai = RetrieveAction(fi, A); /* 搜索功能菜单项 fi 对应的 Action 集合 Ai */

(2) a = GetCurrenttAction(Ai); /* 获得当前 Action */

Ki = GetKeywords(fi, a); /* 获得 fi, a 对应的关键字集合 Ki */

(3)k = GetCurrentKeyword(Ki); /* 获得当前关键字 k */

D = RetrieveFields(f, a, k, p); /* 根据当前 f, a, k, p 从数据库中搜索出对话窗口对应的数据字段的集合 D */

W = Generate(D); /* 根据当前数据字段集 D 组织生成对应的数据窗口信息 */

while(1)

(4)If (∃r∈ && r.E happen){ /* R 为规则集，若有规则 r 的事件 E 发生时先进行条件计算

①compute r.C;

②excute r.A;} /* 执行相应的动作，更新窗口信息，或退出 */

Else loop waiting event happening; /* 等待规则事件发生 */

4. 动态对话类控件类型判定策略

动态对话类窗口，由窗口标题(显示当前操作命令信息)、状态条(显示当前状况)、命令按钮(通过消息映射函数提供消息、窗口数据处理的通用接口)和数据字段(用于输入用户数据，或显示有关数据信息)组成。数据字段通常以下列交互控件的形式表现在窗口上，一般有：State Box(静态文本框)，Edit Box(文本编辑框)，Check Box(复选框)，Radio Box(单选框)，Single Select ListBox(单选列表框)，Multi－Select ListBox(多选列表单选框)。

根据上述算法(1)，确定了组成窗口的各数据字段的信息，而每个字段可以按照下面策略判定将以何种控件表现在窗口上。

(1)各数据字段的标题显然应以静态文本框控件来表示，若字段的可能值是唯一确定的，即不再需用户输入，则该字段值应以静态文本框控件来表示。

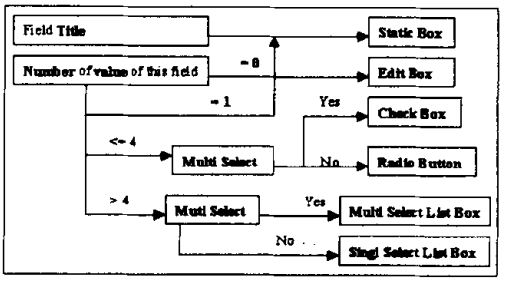

图 2　判定树

(2)否则，按照字段可能值的个数来判定：当字段值的个数为零时即该字段当前工作值为空需要用户输入数

据，即用编辑框控件表示；当值的个数大于 4 时，若允许多选则用复选列表控件表示，否则用单选列表控件表示；当值的个数小于 4 时，若允许多选则用检查框控件表示，否则生成单选按钮控件表示。如图 2 所示：

按照该判定策略，可以将数据字段以适当的交互控件在窗口上表示，供用户选择，或输入。

5. 动态对话类控件布局策略

为了使对话类具有通用性，必须按一定的策略来布局窗口控件，根据上述数学模型和算法，则可以将动态对话类窗口分为六个模块：标题区，动作列表区，关键字段区，动态数据字段区，命令按钮区，状态信息区。它们的布局模式如图 3 所示。

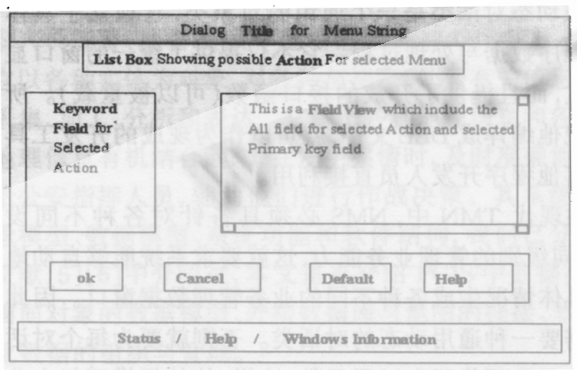

图 3

在上述算法(1)中，(1)由 Ai = RetrieveAction(fi, A) 得到的所有操作 Ai 加入到 Action List Box 中；(2)由 Ki = GetKeywords(fi, a)得到的所有关键字段显示在关键字段区；(3)由 D = RetrieveFields(f, a, k, p)得到所有动态数据字段显示在动态字段区。(4)标题，命令 Button，状态条的位置不变。(5)这里关键字段区，动态字段区的左上角的坐标相对对话窗口是确定的，这样就可以根据字段长度，宽度，字段间距计算出对应区域的大小，当大小超出一定程度时使它们能上下，左右滚动，以保证字段都能正确显示出来。

这样，就可以根据当前查询到的数据字段的多少来合理地构造窗口框架。

6. 动态对话类的设计

动态对话类要具有通用性和开放性就必须将窗口中的具体数据和窗口生成程序、处理程序分开。将所有窗口交互字段信息分解并存储到数据库中，当需要构造或更新窗口对象时，实时地搜索数据库得到新窗口的数据字段信息并生成新窗口。数据库与对话类之间的关系如图 4 所示：(HyperDialog 为动态对话类名)

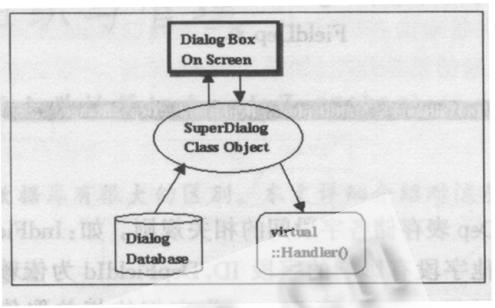

图 4

可以看出，对话类负责提取数据，显示窗体，及缺省的消息处理函数。对话类的设计包括数据库设计，对话类的设计。

(1)动态对话类的数据库设计例子

为了实现动态生成对话窗口，如用以下几个数据表来存储模块信息、对应窗口的字段信息等。这里仅给出主要域的定义。

MenuCmd 表

MenuID	Action Name	OpWinTitle	OperationID	.....

MenuCmd 表存储模块、动作等信息。如：MenuID：应用程序菜单标识号，它对应于一个功能模块；Action Name：与菜单对应的动作。OpWinTitile：对应对话窗口标题。OperationID：标识与 MenuID、ActionName 唯一对应的应用处理号。

FieldFilter 表

OperationID	Keyword	FieldID	.....
			.....

FieldFilter 表存储字段过滤信息。如：Keyword 是与操作号对应的关键字段，FieldID 是由 OperationID，Keyword 决定的字段标识号。

FieldData 表

FieldID	DispName	DataType	Value Range	Default value	Flag	.....

FieldData 表存储字段信息。如：DispName 是字段标

题，DataType 是字段数据类型，ValueRange 是字段值的范围，是字段缺省值，Flag 是字段标记。

FieldDep 表

FieldDep 表存储各字段间的相关规则。如：IndFieldId 为对其他字段有影响的字段 ID，DepFieldId 为依赖字段 ID，IndValRange，DepValRange 为它们的相关取值范围。

(2)动态对话类的设计例子。根据上述算法，实现动态对话类窗口需要定义以下几个辅助对象类：

·HyperDialog Class(动态对话窗口类)，它处理 Action List(动作列表)，Keyword(关键字段)，Command Button(命令按钮)等控件的事件(消息)。另外，根据当前条件，动态的查询数据库、动态更新窗口大小和各控件的位置。

·HFieldView Class(动态视图类)，它是一个可变大小，可上下，左右滚动的视窗类。它负责动态窗口中的动态控件字段的正确显示和管理；计算其中字段的位置大小，使它们恰好对齐。

·HyperField Class(动态字段控件类)，负责管理窗口中的数据字段控件的创建，撤消，与用户交互，数据有效检验，数据交换。

·HFieldData Class(字段数据类)，它存储了字段的显示信息和交互信息。

·HDatabase Class(辅助数据库类)，动态对话类提供了一种机制，超脱于具体的应用程序及具体的应用窗口。因此，它需要辅助数据库来存放应用程序信息，应用窗口信息。所以要一数据库类来读取库中的有关数据信息。

限于篇幅下面简要介绍各类的主要数据成员，如：

```
Class AFX _ CLASS _ EXPORT HyperField: public CObject {
/ * 该字段的超始控件 ID，该字段的结束控件 ID，控件值被改变标记，父窗口指针，字段标题，各种交互控件对象指针，字段变量取值范围，当前字段控件类型，字段有关属性，… * /};
Class AFX _ CLASS _ EXPORT HFieldView : public CScrollView{
/ * 字段对象列表，字段数，…};
Class AFX _ CLASS _ EXPORT HFieldData{
/ * 字段控件标识号，字段标题，字段数据类型，字段的宽度，缺省取值，存储数据的数组，… * /}
Class AFX _ CLASS _ EXPORT HyperDialog: public CDialog{
/ * 父窗口类指针，标题，动作列表，关键字区域的高度和宽度，关键字列表，动态视图的坐标，当前动作，动态视图对象，动态字段控件，… * /};
```

7. 动态对话类的应用

该动态对话类是一个通用的对象类，它独立于具体应用程序数据与处理过程。它不仅提供了统一的窗口显示模式，而且提供了开放的接口函数(可以被重载)。所以若把他编译成 DLL 文件，就可以作为现成的开发工具箱被其他程序开发人员直接利用。

在现代 TMN 中，NMS 必须具备针对各种不同设备、不同级别的管理业务能力，这就要求系统能够自动地根据具体情况生成各种不同的业务管理数据窗口。因此迫切需要一种通用动态的对话类。否则就要为每个对话窗口编写界面代码和过程函数；这样，软件规模不仅十分庞大，而且代码利用率较低，不利于代码维护。为此，将所有窗口数据分解到数据库中，再由该动态对话类对象来自动地生成、管理窗口。笔者在参与国外全球电信管理网系统开发中提出并运用了该动态对话类，明显降低了代码的规模，程序具有很强的灵活性、稳定性、开放性和可维护性，达到了十分理想的效果。

参考文献

[1] Microsoft Foundation Class Libaray. 微软公司

[2] 陈增荣，软件开发方法。复旦大学出版社。

[3] 姜跃平，ECA 规则的模型和行为特定理论。软件学报，1997：Vol 3.

[4] Geoff Caryer. TMN 管理层和管理业务概况的分析. 现代电子科技，1997，Vol9.

(来稿时间：1998 年 11 月)