

Realization of Multi-Object Monitor-Control System Based on COM

基于COM技术的多类对象的监控系统的实现

1 引言

Microsoft 组件对象模型 (Component Object Model, 即 COM) 是开发软件组件的一种方法。在过去的Windows编程岁月里, 一个应用程序通常是由单个的二进制文件组成的。当编译生成此应用程序之后, 在对下一版本重新编译并发行新生成的版本之前, 应用程序一般不会发生任何变化。操作系统、硬件以及客户需求的改变必须等到整个应用程序被重新编译之后才能够得以认可。应用程序在发展历程中是一块块的岩石。目前, 这种状况已经发生了变化, 开发人员找到一种方法, 使应用程序在发行之后不保持完全静止的状态, 以便给已经发行的软件不断的注入新的活力。这种解决方案是将单个应用程序分隔成多个独立部分, 即组件。

在传统的软件工程方法中, 应用程序可以认为是一个单模应用程序, 而在基于COM技术的软件工程方法中, 单模应用程序被一个个组件所取代, 成为组件应用程序, 可以看出: 组件实际上是一些小的二进制可执行程序, 它们可以给应用程序、操作系统以及其他组件提供服务。开发自定义的COM组件就如同开发动态的、面向对象的API(Application Programming

摘要: 为了解决多类对象的监控系统的设计, 本文提出了基于COM技术的解决方案, 并阐述了此方案的设计思路, 同时解决了此方案中的关键性问题: COM技术中多级接口的问题, 最后给出此方案的具体实现。

关键词: COM 接口 ATL

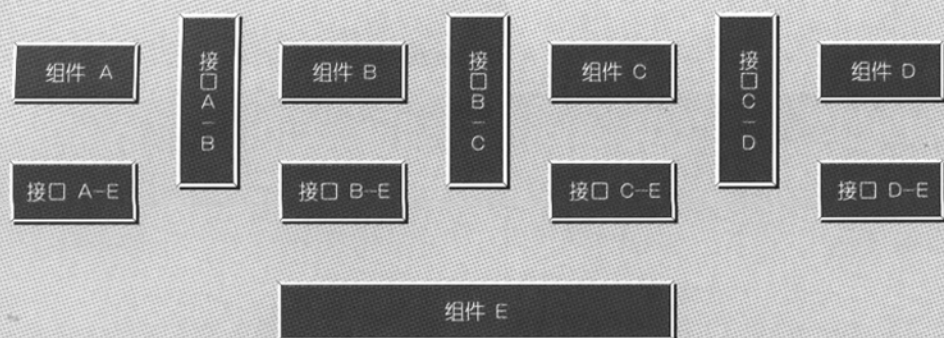


图 1 应用程序组件模型

Interface)。COM以组件对象为中心的设计方法把硬件以芯片为中心的工艺思想恰如其分地融合于软件的面向对象的分析、设计和施工之中,使面向对象的概念和方法从工具语言的层次一下子跃上了系统的应用层,也为 Windows DNA (Distributed interNet Application Architecture) 思想奠定了现实基础。

在COM中接口就是一切。对于客户来说,一个组件就是一个接口集。客户只能通过接口才能同COM组件打交道。从整体上讲,客户对于一个组件可以说是知之甚少的。通常情况下,客户甚至不必知道一个组件所提供的所有接口。可以看出:接口实际上是无缝的被集合在同一名称下的相关方法集。例如:Win32 API就是Windows操作系统的功能接口。采用组件软件技术,不仅操作系统可以使接口可用,而且普通人创建的软件组件也可用。接口是软件组织和其客户之间严格类型化的契约,它的期望行为和期望响应也是相当清晰的,使用组件使得接口可以向程序员和设计师提供具体的实体,以供使用。

在用COM技术开发组件软件时,一个应用程序的组件模型如图1所示:

从图1可以看出:已经定义了组件A、组件B、组件C、组件D、组件E、接口A-B、接口A-E、接口B-C、接口B-E、接口C-D、接口C-E、接口D-E。一般情况下,在此组件模型下开发所需要的组件软件已经没有什么问题。但是,在具体的程序设计时,程序员有可能遇到如下的

问题:在组件A中,需要频繁的与组件D交换数据,利用已定义好的接口,我们目前有两种方法在组件A中访问组件D的数据:

(1) 在接口A-B内调用接口B-C,然后在接口B-C内调用接口C-D,最后才能访问组件D内的数据。

(2) 在接口A-E内调用接口D-E,然后访问组件D内的数据。

这两种方法虽然能够实现在组件A中访问组件D内的数据,但是在此组件模型中,两个缺点也是显而易见的。第一点:程序运行效率太低;第二点:程序员在开发此组件软件的过程中,需要查找多个接口才能访问到组件D的数据,这样,就特别不方便。于是,能不能在组件A中直接访问组件D的数据,就成为COM技术中迫切需要解决的多级接口问题。

在以往的多类对象的监控系统的设计中,由于受到开发工具的限制,各个监控对象间相互牵扯,给程序员的设计带来很大的不便,特别是对系统的拓展,尤其不利。而采用解决了多级接口的COM技术进行开发的话,这个问题就能得到很好的解决,因为在COM技术中,程序员之间的开发互不牵扯,一个程序员所编写的代码对另一个程序员来说,就是一个接口。这样,每一个监控对象用不同接口或是用同一接口的不同属性来实现的话,就给多类对象监控系统的开发带来很好的设计思路。同时,基于COM技术的多类对象的监控系统,很方便就能实现

组件的扩充,为监控系统的拓展带来了很大的便利,使软件的智能化水平大大提高。

2 多类对象监控系统的实现

在图2所示的铁路沿线无人值守机房监控系统中,被监控的机房有25个,被监控的状态有:烟雾、火、温度、湿度、水等,每一个状态我们看成一个监控对象。其中每个机房都有编号、名称、多个Program(监测内容),每个Program又有一个编号,多个监控状态。如果想在一次接口调用中访问任意一个监控机房的所有监控状态信息,我们可以把此监控系统设计为如下的组件模型如图3所示:

在开发这个监控系统时,选用的开发平台为Microsoft Windows 2000, GUI(Graphical User Interface)端采用的开发工具是Microsoft Visual Basic 6.0, COM采用的开发工具是Microsoft Visual C++ 6.0中的ATL(Active Template Library)模板。

2.1 多类对象的监控系统的接口设计

为了在GUI中,通过一次接口(接口1)调用访问监控机房的所有监控信息,即要在GUI中直接访问到监控状态,我们按如下方法进行接口定义和程序设计。

(1) 用ATL模板生成Project Mon-Con-System.dsw,在Server Type选择Service [EXE];

(2) 在Project中定义两级接口 Mon-Con-Lab、Mon-con-Status;接口 Mon-Con-Lab 所指的是GUI与监控机房集合之间的接口,接口 Mon-con-Status 所指的是GUI与监控机房集合与

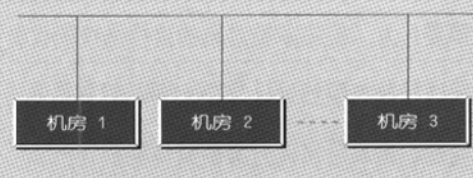


图 2 监控系统示意图

监控状态之间的接口;

(3) 在接口 Mon-Con-Lab 中添加方法
GetStatus ([out,retval] Imon-con-status **ppVal);

(4) 在 Cmon-Con-Houses 类的实现中添加如下代码:

```
STDMETHODIMP Cmon-Con-System:: GetStatus
( Imon-con-status **ppVal)
{
// TODO: Add your implementation code here
|
CComObject<Cmon-Con-Status>* p = new CCo-
mObject<Cmon-con-Status>; p->QueryInterface
(IID-Imon-Con-Status, (void**)ppVal);
return S_OK;
}
```

2.2 多类对象的监控系统的 GUI 设计

在多级接口的问题没有解决之前, 我们对 GUI 设计, 采用的方法是: 设置一个 VARIANT 型变量, 通过一次接口 (接口 1) 调用, 返回的是关于监控机房的 Collection, 其中的每个 Item 代表了一个监控机房。

例如: 我们想得到编号为 1 的监控机房中关于烟雾的状态。

```
Private Sub ReturnStatus_Click()
```

```
    '定义一个新的 COM 服务
```

```
    Dim a as new MonitorCreator
```

```
    '调用接口, 此接口代码内没有添加解决多级接口的代码
```



图 3 监控系统组件模型示意图

```
set coll=a. GetStatus
```

```
'通过接口调用, 访问编号为 1 的监控机房中烟
雾状态
```

```
    'text1.text=coll.Item(1).FogStatus
```

```
End Sub
```

其中, FogStatus 部分不能自动弹出, 要想访问编号为 1 的监控机房中烟雾状态, 还必需再调用一次接口 2, 在监控状态的集合里查找到关于编号为 1 的监控机房的烟雾状态, 返回给 GUI, 这样在使用此接口时, 特别不方便, 但多级接口的问题解决后, 程序员就能很方便的访问监控机房各个监控状态的信息。

例如: 在多级接口问题解决后, 我们想得到编号为 1 的监控机房中的各个监控状态。

```
Private Sub ReturnStatus_Click()
```

```
    '定义一个新的 COM 服务
```

```
    Dim a as new MonitorCreator
```

```
    '调用接口, 此接口代码内添加解决多级接口的代码
```

```
set coll=a. GetStatus
```

```
'通过接口调用, 访问编号为 1 的监控机房中烟
```

```
雾状态
```

```
text1.text=coll.Item(1).FogStatus
```

```
text2.text=coll.Item(1).FireStatus
```

```
End Sub
```

这样就非常方便的实现了在 GUI 中访问各个监控状态的信息。

3 结束语

在开发多类对象的监控系统中, 提出了基于 COM 技术的解决方案, 同时在 COM 技术中, 解决了其中的关键性问题: 多级接口的问题。总的来说, 在开发多类对象的监控系统中, 使用基于 COM 技术的软件工程方法较传统的软件工程方法有了很大的进步, 利用这种开发模式, 很好的解决了在传统的软件工程方法中, 程序员之间, 数据相互牵扯的矛盾。这样, 我们就可以根据用户的需求组成开发小组, 并行、独立地进行开发, 这样不仅可提高效率, 而且使所开发的系统更加稳定、可靠。 ■

参 考 文 献

- 1 David J Kruglinski 著, 希望创作室译, Visual C++ 技术内幕, 北京希望电子出版社。
- 2 Dale Rogerson 著, 杨秀章等译, COM 技术内幕, 清华大学出版社。
- 3 Henry Eddon 著, 希望创作室译, Com+ Base Services 组件编程技术内幕, 北京希望电子出版社。

