

虚拟仪器的面向对象开发技术

吉国力, 杨烜会, 陈 伟, 龚荣盛, 方 翔, 李海良

(厦门大学自动化系, 福建 厦门 361005)

摘要: 对虚拟仪器软件的面向对象开发技术进行了深入的讨论 针对实际应用, 遵循 VPP 体系规范, 围绕虚拟仪器对象的构建和虚拟仪器对象间通讯两项关键技术, 详尽介绍了用 Visual C++ 开发虚拟仪器软件的实用技术

关键词: 虚拟仪器; 面向对象; VPP 体系规范; Visual C++

中图分类号: TP 311.52

文献标识码: A

1 虚拟仪器与面向对象技术

虚拟仪器(Virtual Instrument, VI)是数字化测量技术和计算机技术结合的产物,它是在微机中加入必要的组件(如通用数据采集卡),配备灵活的控制软件和图形化的虚拟面板,来实现测量功能 虚拟仪器的最大特点是其开放性,即用户只要加入一个新的软件模块,就可以构成新的测试系统 在虚拟仪器中,硬件是用来解决测量信号的输入/输出,而软件是整个仪器功能的关键 尽管“模块化”设计是实现虚拟仪器的指导思想,但传统的结构化技术已无法适应虚拟仪器中大量同构部件的组装和复杂并行的交互控制 由此,我们提出了面向对象的解决方案

面向对象技术是从现实世界中的客观事物(对象)出发来构造软件系统 它力图从同构的事物中抽象出对象模板,通过继承形成对象实例,把事物的属性(数据)与功能(方法)封装成整体来实现模块的独立性,用动态绑定的消息通讯来组织对象间的相互作用,使各种对象构成一个有机的系统

虚拟仪器的面向对象开发技术主要是采用抽象、封装、继承等办法来实现各种虚拟仪器部件,并用消息机制来实现虚拟仪器部件间复杂的交互作用和通讯

2 虚拟仪器 VPP 体系规范与面向对象语言

为解决虚拟仪器软件结构的统一性与开放性问题, VXI 总线联合会 1993 年提出了 VPP

收稿日期: 2000-12-06

基金项目: 福建省重点科技项目(2001H020)、福建省自然科学基金(E0110007)和厦门大学科研基金(20001601)资助项目

作者简介: 吉国力(1960-),男,副教授

(VXI Plug&Play)^[1] 体系规范并成为业界标准 VPP 体系规范分为三层(图 1), 即仪器应用程序层、仪器驱动程序层和输入/输出接口层。虚拟仪器应用程序利用微机丰富的交互手段形成控制流, 在仪器驱动程序的帮助下操纵输入/输出接口, 从采样卡获取测量数据并进行处理。VPP 体系规范有效地解决了软件的结构问题, 并为对象的标识提供了切入点, 近年来框架(Framework)和设计模式(Design Pattern)等新的面向对象技术则为实现灵活开放的虚拟仪器框架提供了更为有效的解决手段。目前, 遵循 VPP 体系规范, 采用面向对象技术来开发虚拟仪器已成为主流。

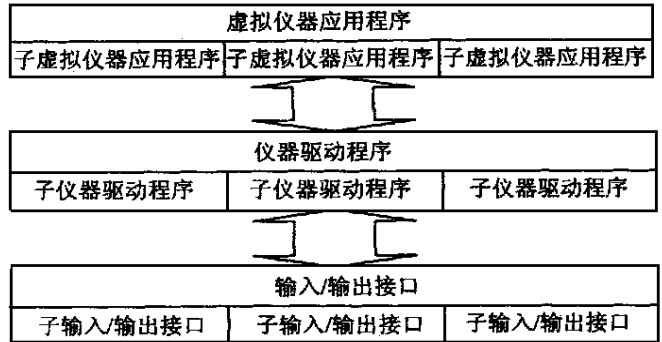


图 1 虚拟仪器 VPP 体系

Fig 1 VPP architecture of virtual instrument

考虑程序的运行速度和底层集成能力, Microsoft 公司的 Visual C++ 是一种理想的虚拟仪器面向对象开发工具。NuMega 公司的 DriverStudio 提供了面向对象的驱动程序开发工具。下面将以 Visual C++ 为例, 结合 VPP 体系规范, 对使用面向对象方法开发虚拟仪器软件进行讨论。

3 虚拟仪器应用程序开发技术

虚拟仪器应用程序中有两类中心对象, 作为交互界中心的虚拟面板和作为数据处理中心的测量数据。虚拟仪器作为子虚拟仪器的集合, 既要提供支撑各个子虚拟仪器的水平框架, 又要提供支撑子虚拟仪器内部各个部件的垂直框架。

在 Visual C++ 中有多种方式可以实现这样的支撑。对话框应用程序(Dialog-Based Application)就是一个简洁而高效的方案。以对话框应用程序的缺省对话框为母板, 用 CDialog 类无模式对话框子面板封装子虚拟仪器面板, 再把测量数据类作为子面板对话框对象的主要属性, 形成子虚拟仪器应用程序对象, 这样就实现了水平和垂直的集成。虚拟仪器对象模型与对应用程序之间的映射关系如图 2 所示。

虚拟面板控件化是实现虚拟仪器“用户设计、用户组装”的基础。Visual C++ 的对话框是以对话框模板为界面, 以 CDialog 为基类的一种容器。对话框模板是交互部件的布局, 可以“所见即所得”(WYSIWY)地编辑, 每个控件都由相应的控件类来管理, 例如用按钮控件(CButton 类)实现仪器面板中的按钮, 用进度条控件(CProgressCtrl 类)实现仪器面板中的柱状指示器等。实际编程时更多地使用现有的或派生的第三方 ActiveX 仪器面板专用控件, 如示波器、度盘等控件来实现复杂交互。编程时, 应坚持面向对象技术的重用观点, 尽量利用现成的控件。

为方便起见, 本文不严格区分“MFC 类库”和“Visual C++ 开发环境”, 而用“Visual C++”来统一表述。

仪器的本质是数据的获取和处理 以面向对象的观点, 数据处理和表达具有共性, 都是对数据的操作 这里, 采用 COBJect 类来派生一个测量数据类, 将采样数据作为测量数据类的主要属性, 将各种测量采样方法、数

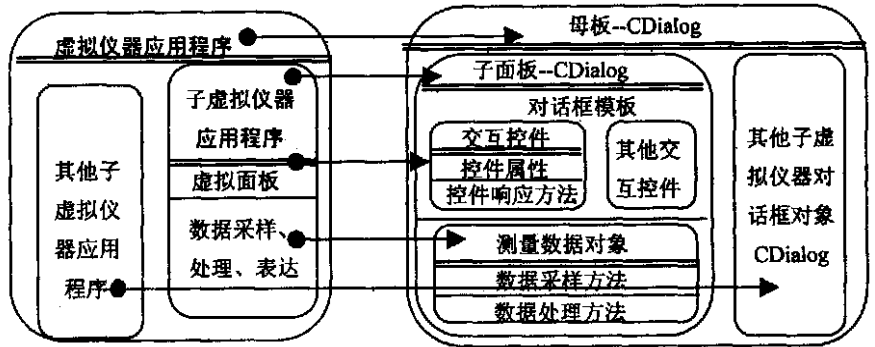


图 2 虚拟仪器应用程序对象模型及其对话框应用程序实现之间的映射

Fig 2 Mapping between object model of virtual instrument and its dialog based application implementation

据处理方法、数据表现方法都作为测量数据类的接口方法 使用这种方法并结合设计模式和多态性有助于接口的规范化和自由扩展

构造测量数据类, 一般直接封装结构和数组, 也有使用模板类的, 如 CA rray 类, CL ist 等, 它们支持复杂类的动态添加 由于测量的数据量一般都比较比较大, 全局对象和共享内存文件也是常用的办法

测量数据可以保存和调入是虚拟仪器的另一个特色, 用 V isual C+ + 的 CFile 类和 CA rchiv 类配合可方便地将测量数据对象串行化, 以流的方式存取 V isual C+ + 还有强大的 DAO、ODBC、ADO 类支持数据库管理测量结果

数据处理、表达流程既可直接引用数据操作方法形成处理序列, 也可用事件响应链形成处理序列 使用事件响应链既便于组合, 又便于通过消息广播机制支持并行处理 事件响应技术是消息驱动的, 一般要使用自定义消息技术 消息发送有两种方式: SendM essage 和 Pos tM essage 还可以用系统注册消息 RegisterW indowM essage 供多个虚拟仪器协作

通过响应用户在虚拟面板控件上的交互事件可以形成控制指令, 测量控制动作则放在面板控件的响应函数中来实现 以消息机制或其他通讯手段触发测量数据对象的采样方法, 去操作虚拟仪器驱动程序对象来获取测量数据; 测量数据到达后, 经过数据处理、表达序列最终呈现给用户, 这就是虚拟仪器应用程序的工作流程 它是通过虚拟面板、测量数据两个中心对象以消息作用网为媒介来实现的

虚拟仪器的交互操作非常自由, 而且一些操作可以并行, 例如数据采样和数据处理并行, 因此控制流程会比较复杂, 有时需要用状态转换图来分析 并行任务可用 CW inThread 类线程来实现 界面元素之间的制约关系可以用W in32 A PI函数LoadCurreor 调用等待光标来暂时封锁其他交互操作, 也可以用 CW nd: : EnableW indow, CW nd: : ShowW indow 设置相应界面控件来挂起部分交互操作

CD ialog 类作为容器, 具有良好的分治及组态效果 虚拟仪器应用程序组织方式还有单文档界面(SD I)和多文档界面(MD I)两类视/文档(V iew /Doc)应用程序结构 为便于虚拟面板设计, 采用视/文档结构时, 一般选 CFomV iew 作V iew 的基类, 这样便于对话框模板编辑和



利用视/文档结构的文档存储特性 无论如何实现, 我们强调: 要坚持每个仪器模块对应一个测量数据类, 要遵循VPP规范的界面标准, 要用好仪器人机工力学的有关原理

以下给出了一个虚拟仪器(图3)的开发实例 在该仪器的开发实现中, 采用了面向对象开发技术, 如: 操作界面采用了对话框程序结构, 功能组切换采用CTableCtrl控件, 测量数据交换采用全局对象, 对象间通讯采用自定义消息等

4 虚拟仪器驱动程序和输入/输出接口开发技术

仪器驱动程序是虚拟仪器应用程序与输入/输出接口的桥梁, 其作用主要是提供采样硬件的抽象操作集合

抽象操作集合的实现思路是: 利用面向对象技术的抽象与封装机制, 把输入/输出接口包装成一个受控的数据源, 来填装虚拟仪器应用程序中的测量数据对象 实现方法是利用Visual C++的COBject基类形成一个基类, 利用串行化方法COBject::Serialize()来提供流式数据, 然后各个子仪器驱动程序继承并重载该类的Serialize()方法

硬件驱动一般有两种方式, 轮询和事件触发 轮询方式由于要等待, 系统资源浪费比较大而事件触发方式类似于中断方式, 是实现虚拟仪器驱动程序功能体经常采取的方式 实际上, 目前开发驱动程序的流行工具大多采用面向对象技术, 比如DriverStudio 实现上, 还得考虑: 数据采样过程出现不可预见情形的超时保护措施; 在定时方面, 采用WM_TIMER事件响应函数OnTimer, Win32 API多媒体时钟函数timeSetEvent, QueryPerformanceCounter等; 若要实现更高精度的定时, 采用CMOS实时时钟RTC等

Visual C++有CSocket类支持网路通讯, 由此, 可以把网络数据采集集成进来, 开发网络仪器驱动程序扩充虚拟仪器应用程序 基于DCOM和CORBA模型, 来实现分布式测量提供解决方案

虚拟仪器的部分自适应测试智能, 如: 自诊断、自动增益切换、极性判断等功能, 可用仪器驱动对象的私有方法来实现

关于提供抽象操作集合的方式, VPP体系规范建议两种接口: 图形交互式接口和程序开发式接口 像图形交互式接口MSCommCtrl串口控件就是一个很好的图形交互式串口仪器驱动程序 程序式接口的仪器驱动层可以以库(静态的、动态的)的方式提供, 也可以以驱动程序的方式提供, Visual C++对多种驱动开发工具都提供了很好的支持 对于接口形式, VPP体系规范都要求设计和实现形式一致, 结构模块化, 并要提供源码, 以便用户理解和使用

在编程实现时, 往往要根据硬件情况和规模, 折衷考虑虚拟仪器驱动程序与输入/输出接口功能关系 实现思路同仪器驱动程序的开发一样, 即把硬件包装成实体对象来使用

虚拟仪器软件系统调试是一个复杂烦躁的过程, 而采用面向对象的模块化非常有利于渐增式的组装调试 在按VPP体系规范开发仪器软件的各个层次时, 都应适当留出一些调试接口; 有效利用消息发送方式对任何层次进行模块激活; 利用C++语言提供的对象构造函数、自动执行及多态性, 有效处理软件开发过程中的部件初始化问题, 尽可能简化软件开发过程

5 结 论

由以上的讨论可知, 通过继承或派生虚拟仪器软件对象, 可以快速地构造出虚拟仪器软件部件; 利用框架技术, 可以有效地组织管理虚拟仪器软件对象; 依靠消息映射和数据交换机制,

可以建立灵活安全的对象之间通讯框架

总之,采用面向对象技术开发虚拟仪器软件,有利于开发易组装、易维护的虚拟仪器产品对模块安全性、快速原型化、高可重用性、接口统一性、并发流程控制都可得以改善或提高,而这些特性是传统的结构化编程难以得到的

参考文献:

- [1] VPP-2 System Frameworks Specification[S]. VXI Plug&Play System Alliance, 1996
- [2] 周泓,汪乐宇,陈祥献 虚拟仪器系统软件结构的设计[J]. 计算机自动测量与控制, 2000, 8(1): 21- 24
- [3] 邵维忠,杨芙清 面向对象的系统分析[M]. 北京:清华大学出版社, 1998
- [4] 杨芙清 面向对象程序设计[M]. 北京:北京大学出版社, 1994
- [5] David J. Kruglinski Inside Visual C++ , 4th Edition[M]. U. S. A: Microsoft Press, 1997.

The Object-oriented Development of Virtual Instrument

J I Guo-li, YANG Xuan-hui, CHEN wei,

GON G Rong-sheng, FANG Xiang, L I Hai-liang

(Dept. of Automation, Xiamen Univ., Xiamen 361005, China)

Abstract: In this paper, object-oriented(OO) techniques for developing virtual instrument (V I) software are discussed in depth. Practical skills, which are used to implement virtual instruments with Visual C++ on the basis of the VXIbus Plugs&Play (VPP) architecture specifications, are described focusing on the constructing of virtual instrument objects and the communicating among them.

Key words: virtual instrument (V I); object-oriented technique; VPP specifications; Visual C++