

文章编号: 1673-9965(2008)01-049-05

基于组件与 Petri 网技术的虚拟仪器的研究*

杨盛泉, 刘萍萍, 李宝敏

(西安工业大学 计算机科学与工程学院, 西安 710032)

摘要: 为了建立可以与任何通用软件系统无缝集成的虚拟仪器模型, 提出了运用组件建立多线程共享缓冲区的通用虚拟仪器架构, 同时引进 Petri 网技术来分析和控制组件内部多线程运行中产生的并发与冲突、活性、安全性等. 实验表明, 该技术由于不受平台和语言环境的限制, 能够增强虚拟仪器组件与应用程序开发的独立性和使用的安全性, 能够大大地拓展虚拟仪器的应用范围.

关键词: 虚拟仪器; 组件; 线程; Petri 网

中图分类号: TP311 **文献标志码:** A

自从美国 National Instruments 公司八十年代提出虚拟仪器概念以来, 虚拟仪器技术得到大力的发展, 正在趋于成熟和完善并且在各个领域里得到广泛的应用. 应用虚拟仪器的目的是为了达到缩短控制系统的开发时间、提高生产质量并大幅度地降低生产成本. 虚拟仪器系统的基本构架包括功能强大又高效的软件、模块化的测量硬件, 它是以软件为核心, 由计算机和必要的硬件组建起来的测试系统或仪器系统. 虚拟仪器可以利用一台计算机集成多个仪器的功能, 从而节约硬件空间和成本^[1].

目前, 市面上技术成熟的虚拟仪器软件开发平台相当多, 主要有 LabWindows/ CVI, LabView, HP VEE 等^[1]. 但它们有一个共同缺点: 它们实现的虚拟仪器很难被分离出来在通用编程语言(如: VC++, VB, Delphi 等)环境下得到重复使用, 其代码的可移植性极差. 然而, 基于组件和 Petri 网技术实现的虚拟仪器, 可以实现和任何通用软件系统无缝集成. 由于不受平台和语言环境的限制, 大大地拓展了虚拟仪器的应用范围.

1 组件概述以及虚拟仪器组件模型

1.1 组件概述

组件技术来源于面向对象技术, 但又不同于对

象. 一个对象是一部分源代码, 对象可以实现源代码级的复用, 对外是不可见的. 而组件不一定是需要经过编译才能使用的一部分源代码, 可以是能实际运行的软件模块. 组件通过提供开放的接口来实现二进制级的复用. 组件是一种可以复用的程序段, 且可以为二进制形式. 组件的范围小到图形界面中的按钮, 大到一个具有相当功能的文本编辑器. 一个在应用程序中的组件对于用户可以是可见的, 也可以是不可见的. 组件独立于开发组件的语言, 而面向应用程序, 只规定组件的外在表现形式, 而不关心其内部实现方法. 组件具有可靠、模块化、可复用、可扩展等特点. 区分组件和其他先前的技术有四个原则^[2], 即封装 (Encapsulation)、多态性 (Polymorphism)、后期连接 (Late Binding) 和安全性 (Safety). 这个列表与面向对象是重复的, 除了它删除了继承 (Inheritance) 这个重点.

1.2 虚拟仪器组件模型

虚拟仪器组件模型的形式可以定义为一个五元组: $VIC = (P, M, E, I, D)$, 其中 P 表示虚拟仪器组件的属性 (Property) 的集合, M 为虚拟仪器组件所有方法 (Method) 的集合, E 为虚拟仪器组件所能发生的事件 (Event) 的集合, I 为虚拟仪器组件提供给用户操作接口 (Interface) 的集合, D 为虚拟

* 收稿日期: 2007-10-19

作者简介: 杨盛泉 (1974-), 男, 西安工业大学讲师, 主要研究方向为 MIS 系统、智能控制、软件工程. E-mail: xaitysq@163.com.

仪器组件模型、输出的驱动程序集合。

采用虚拟仪器组件开发的应用软件系统是通过虚拟仪器组件模型间相互提出请求及提高服务来协同工作,各虚拟仪器组件模型需要具有良好的接插特性^[3]。虚拟仪器组件模型技术可以提高开发速度,由于大量使用开发商已经编制好的组件模型,减少了用户开发的工作量,使开发周期大大缩短^[4]。虚拟仪器组件模型技术可以降低开发成本,由于用户进行软件开发工作量大大减少,因此开发成本也相应降低,通常,组件模型的费用相对于传统方法开发的成本来说是微不足道的。

2 虚拟仪器组件的体系结构及其 Petri 网分析模型

2.1 组件的体系结构

运用组件建立多线程共享缓冲区的通用虚拟仪器,每个线程都有它自己的运行内存、句柄和系统资源。通用虚拟仪器组件由5个线程构成:调度控制管理线程、外设通讯线程、数据处理线程、数据存取线程和数据显示线程,其体系结构如图1所示。

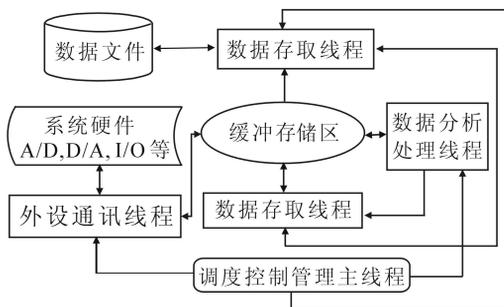


图1 多线程共享缓冲区的虚拟仪器体系结构

Fig. 1 The system structure of virtual instrument with multi-thread sharing buffer

该虚拟仪器组件中的各个线程具体含义如下:

1) 调度控制管理线程为虚拟仪器组件的主控制线程,负责其他4个工作线程的并发、同步调度管理工作。它随着组件的创建而产生,随着组件被销毁而消亡。

2) 外设通讯线程控制配置给硬件,正常情况下它是一个自治系统,即它自动在后台和硬件完成常规的通讯工作,同时它也接受外界采集测试数据和控制硬件的指令。

3) 数据处理线程负责对测试数据的分析处理,如剔除干扰假象数据、计算数据均值和峰度等。

4) 数据存取线程负责与外设磁盘的数据交换工作,如原始测试数据的存取、处理结果的存取、历史数据的查阅等。

5) 数据显示线程则把测得的数据或处理后的结果以图形、表格等形式显示出来,还可以显示历史保存在文件中的测试或者分析数据。

各线程之间通过一个共享的缓冲区来传送数据。

多线程共享缓冲区的虚拟仪器结构有如下优点:

1) 元件可重用。所谓元件是指能完成一定测试或处理功能的可执行程序。每台虚拟仪器的4个工作线程分别代表4种类型的仪器元件,这些元件存储在虚拟仪器平台的元件库中作为模板,针对新的运行环境,对其属性稍作修改,就可以被新仪器使用。

2) 有助于实现虚拟仪器平台的开放性。由于这些线程都是独立的,当提供新的外围硬件时,只要修改相应的外设驱动程序,而不必改变高层的应用程序。

3) 采用多线程结构来实现虚拟仪器,可以简化系统软件设计,通过把控制硬件采集数据从数据的分析和处理中分离出来,避免编程的复杂性。

4) 多线程可以更有效地利用CPU处理资源。在高速采集场合,可避免系统忙于采集而不能有效地存储采集数据和响应用户事件。

组件内部的多线程运行时,它们之间肯定存在资源竞争与同步问题,下面就用Petri网建立模型并解决这个问题。

2.2 Petri 网分析模型

Petri网是异步并发系统,没有人为的控制流,直观地表示了非确定性;Petri网自身具备的可运行性方便了系统形式化描述级的模拟,可以用于表达不同抽象级上的系统概念并清楚地描述整个系统的运作过程^[4]。

Petri网是一种使用图形方式对系统进行需求规格说明的技术,用来定义多进程/线程、多任务系统的数学模型,易于描述系统的并发、竞争、同步等特征,并可用于评价和改进系统。Petri网不仅能描述同步模型,更适合于相互独立、协同操作的处理系统^[5-6]。

将图1所示的多线程共享缓冲区的虚拟仪器描述成Petri网,如图2所示。

Petri 网由 4 种不同的元素组成,即库所 (Place,用“O”表示),变迁 (Transition,用“-”表示),连接库所和转移的有向弧及位于库所中令牌 (token,用“·”表示).库所表示系统状态的逻辑描述,转移表示系统中事件或行为的产生过程,输入函数(I)和输出函数(O)分别表示库所和转移之间的连接函数关系,若一个库所被赋予 k 个标记 (k 为非负整数),则说明该库所有 k 个令牌,也称该库所被标记,于是定义标记 Petri 网为一五元组 $PN = (P, T, I, O, M)$, M 是 Petri 网的状态标识集^[7].表 1 详细说明了本模型中库所和变迁的具体含义.

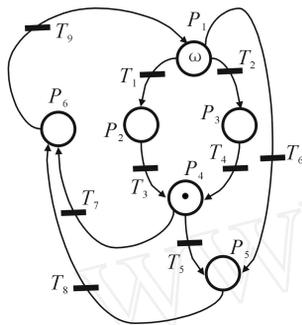


图 2 多线程共享缓冲区的虚拟仪器 Petri 网模型

Fig. 2 The Petri nets model of virtual instrument with multi-thread sharing buffer

表 1 Petri 网模型具体描述含义

Tab. 1 The concrete descriptive meaning of Petri nets

库所	表示状态或行为	变迁	表示命令或事件
P_1	系统处于主控制调度循环	T_1	发送采集数据或控制动作命令
P_2	执行和外设通讯任务	T_2	发送分析处理数据命令
P_3	执行数据分析处理任务	T_3	保存通讯返回数据入缓冲区事件
P_4	执行缓冲数据读或写任务	T_4	保存分析处理输出入缓冲区事件
P_5	执行磁盘文件数据读或写任务	T_5	缓冲区存储满(需存盘)事件
P_6	执行数据显示刷新任务	T_6	查看磁盘历史数据命令
		T_7	显示采集返回或者分析输出事件
		T_8	显示磁盘历史数据事件
		T_9	数据画面显示或者刷新完毕事件

2. 2. 1 模型的并发和冲突分析^[8-9]

由建立的多线程虚拟仪器 Petri 网模型知,多

线程仪器在许多情况下允许仪器内部线程并发运行.比如,对标识 $M_0 = (\cdot, 0, 0, 0, 0, 0)$, 变迁 T_1, T_2, T_6 都是使能的,因为:

$$M_0 [T_1] M_1 = (\cdot, 1, 0, 0, 0, 0)$$

$$M_0 [T_2] M_1 = (\cdot, 0, 1, 0, 0, 0)$$

$$M_0 [T_6] M_1 = (\cdot, 0, 0, 0, 1, 0)$$

其中, $M_i [T] M_j$ 表示变迁 T 的激发(又称点燃),使 Petri 网由标识 M_i 进入标识 M_j .

“允许 T_1, T_2, T_6 并发”的实际含义是虚拟仪器的数据和外设通讯、数据分析处理及历史数据查阅 3 个线程可以并发运行.但是,多线程共享的缓冲区为系统的临界资源,它任何某一时刻只能为某一个线程提供读写服务.从该 Petri 网模型中可以看出,在多线程仪器中不允许同时运行 2 个线程数据读写操作.因为在标识 $M_n = (\cdot, 1, 1, 0, 0, 0)$ 下, T_3, T_4 互相冲突.冲突的实质是资源竞争,在标识 M_n 下, T_3 和 T_4 竞争条件 P_4 的空间资源.

2. 2. 2 资源竞争冲突的解决方案

4 个并发的线程对临界资源的访问必须加以控制才能使得虚拟仪器组件能够正常的运行.援引操作系统中的对进程调度管理的策略理论就可以解决此问题.引入信号量 S (初值为 1).在 S 上可以进行 P 操作和 V 操作.

P(S) 定义如下:

- 1) $S = S - 1$;
- 2) 如果 $S \leq 0$, 则该线程继续进行;
- 3) 如果 $S < 0$, 则该线程被封锁,并插入到信号量 S 的阻塞队列中,系统要重新调度另一线程投入运行.

V(S) 定义如下:

- 1) $S = S + 1$;
- 2) 如果 $S > 0$, 该进程继续运行;
- 3) 如果 $S \leq 0$, 则表示有线程处于阻塞状态,必须从信号量 S 的阻塞队列中唤醒线程,使之进入就绪队列,并继续执行本线程的操作.

每个线程的控制代码类似如下:

Thread Some_Thread_Work(i)

Begin

.....

i 线程常规工作;

.....

P(S); // 申请操作权限

i 线程操作缓冲区;

V(S) ;// 释放操作权限

.....

End;

2.2.3 模型的结构特性分析^[10]

设 $*M_0$ 是从 M_0 开始的可达标识集, 若对每个可达标识 $M_i \in *M_0$, 存在一个从 M_i 开始的包含 T_j 的激发序列 S , 则称变迁 T_j 对初试标识 M_0 是活的. 如果所有的变迁对 M_0 都是活的, 则称该 Petri 网对标识 M_0 是活的. 如图 2 所示的 Petri 网既没有源位置和井位置, 也没有源变迁和井变迁, 不论标识如何发展, 其中的每个变迁都有激发的可能, 因此该 Petri 网是活的. 又因为初始标识 M_0 存在:

$M_0 [T_1, T_3, T_5, T_8, T_9] M_0$

$M_0 [T_1, T_3, T_7, T_9] M_0$

$M_0 [T_2, T_4, T_7, T_9] M_0$

$M_0 [T_2, T_4, T_5, T_8, T_9] M_0$

$M_0 [T_6, T_8, T_9] M_0$

所以图 2 所示的 Petri 网是多通路重复的. 多线程 Petri 网是活的和重复的, 说明多线程虚拟仪器不会发生运行死锁现象.

3 虚拟仪器组件的实现

目前组件的标准化主要有两大阵营: 对象管理组织 (OMG) 的 CORBA 和 Microsoft 的 COM/DCOM/COM+, 它们在组件的结构以及组件之间的通信方面提供了不同的机制. 因此在将组件技术应用于虚拟仪器的时候, 可以根据实际的应用方向选择合适的标准来构造虚拟仪器组件.

运用 VCL 组件技术和 Petri 网技术在 Windows XP 和 Borland Delphi 7.0 的环境下开发了一个虚拟工业温度控制仪表组件, 运行实例如图 3 所示.

VCL 虚拟温度组件主要头代码定义如下:

```
TVIGauge = class(TSoft Gauge) // 组件类定义
private
{ Private declarations }
FPV :real ; // 组件测量温度
FSV :real ; // 组件设定温度
FOP :real ; // 组件输出控制
Fevent : TEvent ; // 用户事件
FDriver : TDriver ; // 组件设备硬件驱动
```

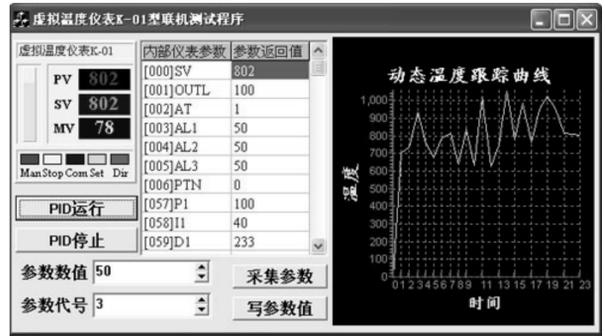


图 3 虚拟温度仪表组件开发实际运行画面

Fig.3 The actual run window of development of the virtual temperature instrument component

```
FInput ValueChannel : TAD_ IOValue ; // 硬件数据采集
```

```
FOutput ValueChannel : TDA_ IOValue ; // 硬件数据输出
```

.....

本虚拟仪器组件可以在任何语言下重复使用, 该组件易于升级和继承, 当硬件驱动升级时, 可以单独对其进行修改, 只要对外的接口保持不变, 就能平滑地将新的组件替代原来的组件.

4 结论

基于组件与 Petri 网技术的研究虚拟仪器的方法将复杂的问题分解得简单化, 具有很强的通用性和科学严谨性, 为虚拟仪器研究与开发工作提供了较好的模型和方法, 具有较高的实际应用价值和光明的研究前景.

参考文献:

[1] 袁渊, 古军, 习友宝, 等. 虚拟仪器基础教程[M]. 成都: 电子科技大学出版社, 2002.
YUAN Yuan, GU Jun, XI You-bao, et al. Basic Tutorial of Virtual Instrument [M]. Chengdu: Electronic Science University Press, 2002. (in Chinese)

[2] Antonio Brogi, Carlos Canal, Ernesto Pimentel. Measuring Component Adaptation[C]// Coordination Models and Languages (Coordination '04). Berlin: LNCS, 2004: 72.

[3] 王见, 秦树人, 柏林. 网络虚拟仪器的开发研究[J]. 重庆大学学报: 自然科学版, 2004, 27(11): 137.
WANG Jian, QIN Shu-ren, BO Lin. Research and Development of the Network Virtual Instrument [J]. Journal of Chongqing University: Natural Science E-

- dition, 2004, 27(11):137. (in Chinese)
- [4] 曾雪辉, 杨路明. 虚拟实验室中用户注册组件的设计与实现[J]. 计算机应用, 2007, 27(1):189.
ZENG Xue-hui, YANG Lu-ming. Design and Implementation of Registering Component by User in Virtual Laboratory[J]. Computer Applications, 2007, 27(1):189. (in Chinese)
- [5] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社, 2005.
YUAN Chong-yi. The Principle and Application of Petri Nets[M]. Beijing: Publishing House of Electronics Industry, 2005. (in Chinese)
- [6] 彭高亮, 刘文剑, 张旭堂. 基于 Petri 网的虚拟装配环境建模及其应用[J]. 计算机集成制造系统, 2007, 13(8):654.
PENG Gao-liang, LIU Wen-jian, ZHANG Xu-tang. Modeling & Application of Virtual Assembly Environment Based on Petri Nets[J]. Computer Integrated Manufacturing Systems, 2007, 13(8):654. (in Chinese)
- [7] 洪流, 马巧云, 陈学广. 基于 Petri Nets 的协调机制研究[J]. 计算机应用研究, 2007, 24(4):26.
HONG Liu, MA Qiao-yun, CHEN Xue-guan. Study of Coordination Mechanisms Based on Petri Nets[J]. Application Research of Computers, 2007, 24(4):26. (in Chinese)
- [8] Pedrycz W, Camargo H. Fuzzy Timed Petri Nets[J]. Fuzzy Sets and Systems, 2003, 140(2):311.
- [9] Busi N. Analysis Issues in Petri Nets with Inhibitor Arcs[J]. Theoretical Computer Science, 2002, 275(1/2):128.
- [10] DING Zouhua, Bunke H, Kipersztok O, et al. Fuzzy Timed Petri Nets analysis and Implementation1[J]. Mathematical and Computer Modeling, 2006, 43(3/4):390.

Research of Virtual Instrument Based on Component and Petri Nets Technology

YANG Sheng-quan, LIU Ping-ping, LI Bao-min

(School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710032, China)

Abstract: In order to build a virtual instrument model that can be integrated into any general software without gaps, this paper presents a general architecture with multi-thread sharing buffer established by component theory, then analyzes and controls the problems such as concurrence, collision, aliveness and security etc., during the process of the multi-threads running inside the component by importing the Petri nets technology. It can be proved by scientific experiment that because of not being restricted within any platform and language environment, this technology can enhance the independence between application program and VI component, and it also enhances the security when use VI in the application, and it exploits and extends greatly the range of the application of virtual instrument.

Key words: virtual instrument (VI); component; thread; Petri nets

(责任编辑、校对 苗静)