

基于配置文件的虚拟仪器 测试软件设计与实现

刘金宁¹, 孟 晨¹, 崔少辉¹, 陈德祥²

(1. 军械工程学院 导弹工程系, 河北 石家庄 050003; 2. 南京军区 导弹站, 江苏 南京 210028)

摘要: 指出了当前虚拟仪器测试系统中软、硬件集成存在的问题, 如通用能力差, 不是以“测试任务”为中心, 测试平台庞大、复杂等。提出了一种基于配置文件的虚拟仪器测试软件模型, 剖析了模型结构、特点, 论述了模型的具体实现过程。给出了基于该模型的虚拟仪器软件设计方法。基于配置文件的虚拟仪器测试软件实现了软件结构的可配置、软件代码的重用和硬件模块的互换, 简化了系统集成。

关键词: 虚拟仪器; 测试软件; 配置文件; 功能接口

中图分类号: TM 930.9

文献标识码: B

文章编号: 1006-6047(2005)01-0050-04

0 引言

虚拟仪器技术^[1~4]把计算机软、硬件资源有效地结合起来, 从观念上给测试领域带来了一场革命。在虚拟仪器测试系统中, 仪器模块作为测试资源, 为用户提供“测试任务”实现环境; 软件是系统的核心和灵魂, 硬件模块的功能由软件定义, 因此有“软件就是仪器”的说法。

当前, 虚拟仪器测试软件开发中存在很多问题^①。首先, 硬件模块的驱动软件通用能力差, 驱动程序仍以测试硬件自身的“特征应用”为中心, 不同硬件模块的驱动程序在定时、激励、数据结构等功能实现上存在很大差异, 很多相同、相似的功能本可以用相同的代码实现而未实现统一, 这给开发人员使用软件平台集成多个硬件模块带来很多不便, 难以实现“即插即用”。其次, 测试软件的实现仍以“测试资源”为中心, 而不是“测试任务”。由于各厂家在仪器驱动程序设计规范上不统一, 仪器控制仍是测试软件开发的首要工作, 这使得软件集成必须由专门的工程技术人员(既懂测试原理, 又懂仪器控制)完成, 难以实现商业化生产, 而且开发人员的精力很难集中到测试功能的实现上。最后, 某些军用测试平台面向的被测对象 UUT(Units Under Test)种类繁多、规模庞大、结构复杂, 这也给软件开发带来难度, 大大提高了测试系统的研制成本, 浪费了开发时间。

综上所述, 最大限度地提高测试软件设计的灵活性和效能, 实现面向“测试任务”的功能开发, 同时降低软、硬件集成成本, 缩短开发周期是当前虚拟仪器软件开发迫切需要解决的问题。

1 虚拟仪器测试软件模型

配置文件是计算机系统中存储软、硬件信息的

文本, 是 Windows 大厦的关键集成技术之一。它以字段信息的形式实现了 Windows 操作系统的动态配置和软、硬件模块之间的信息交换。基于配置文件的虚拟仪器测试软件模型的测试软件开发把装备工程师、软件工程师和硬件设计人员各自的工作区分开来, 通过配置文件实现了面向“测试任务”的虚拟仪器设计, 提高了软件开发的灵活性和效率。

1.1 虚拟仪器测试软件模型

基于配置文件的虚拟仪器测试软件模型见图 1。

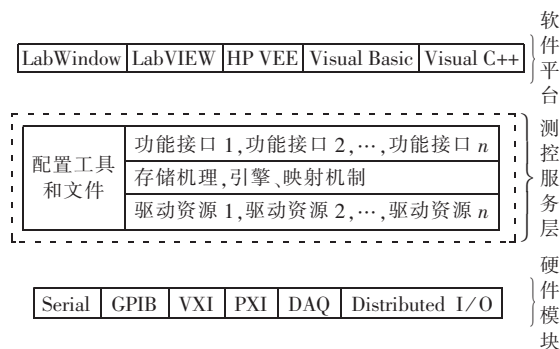


图 1 基于配置文件的虚拟仪器测试软件模型

Fig.1 Virtual instrument test software model based on configuration file

与普通测试软件相比, 除了上层的应用程序开发平台和底层的硬件模块类似外, 该模型增加了以“配置文件”为核心的测控服务层。测控服务层作为中间层为测试系统的软、硬件集成提供服务(如仪器控制、数据分析、显示等), 包括功能接口层、引擎、映射机制层和硬件驱动资源层三部分。其中功能接口层是用户应用程序与硬件模块通信的功能接口, 是实现面向“测试任务”虚拟仪器软件设计的基础。它把测试系统中仪器设备的一些相似操作统一起来, 编制成通用软件模块, 即功能接口函数库提供给上层开发平台调用。比如读取 1 个被测节点的电压

① National Instrument Corporation. Instrumentation Newsletter. Third Quarter 2002, 6-7.

值,使用万用表和 A/D 模块都可以,而具体由哪个仪器执行由硬件电路和配置文件映射信息决定。装备工程师编程时,可以不关心具体的仪器控制过程,而把精力放在面向“测试任务”的测试功能实现上。引擎、映射机制层作为测控服务层的调度中心,规定了测试功能接口与底层硬件驱动程序进行通信的机制,管理 UUT 与测试资源之间的信息映射关系^[6]。硬件驱动资源层是厂家提供的驱动程序库,是硬件模块的上层支持库,它直接与硬件模块通信,提供底层软件控制服务,完成相应的功能调用。测控服务层的三层结构通过配置文件进行信息映射和交换。配置文件作为信息存储中心,完成硬件设备的管理、测试任务的组合设定和测试资源的分配等功能。

1.2 模型实现

基于配置文件的虚拟仪器软件模型的实现主要包括开发测试功能接口函数库和设计配置工具及配置文件两部分。

1.2.1 测试功能接口

基于配置文件的虚拟仪器软件模型中,测试功能接口层连接软件开发平台与底层硬件设备,它把测试软件设计由面向“测试资源”转变到面向 UUT “测试任务”。对于不同的测试需求,只要调用这些功能接口函数就可以完成相应的任务。因此,开发测试资源功能接口函数库是实现该模型的关键环节。由于无论多么复杂的测试系统,其测试功能种类总是有限的,主要包括直流电压、交流电压、电流、频率、时间间隔、时序、状态、频谱、波形、沿宽、信号激励等十余种,所以功能接口数量也是有限的。

以测试直流电压为例,功能接口函数原型如下:
`measure-DC-volt(UUT-pin,DC_Volt……)`
//UUT-pin 是功能接口函数中测试资源和 UUT 沟通的桥梁,是所有映射关系的源头
{
 //函数实现细节,包括存储机理,引擎、映射机制以及具体的仪器驱动程序调用等
}

功能接口函数内部实现如图 2 所示。

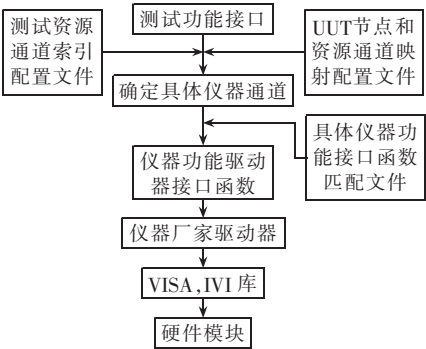


图 2 功能接口函数内部实现框图
Fig.2 Internal implementation of function interface

- 功能接口函数内部实现执行步骤包括:
- a. 根据 UUT-pin 参数、测试通道映射配置文件和测试资源通道索引配置文件确定及 UUT-pin 相连的测试资源通道 Resource-channel;
 - b. 根据测试资源通道 Resource-channel 确定具体仪器功能驱动器(由系统软件工程师在厂家提供的驱动器基础上进一步封装而成);
 - c. 判断该仪器是否具有 measure-DC-volt 功能,如没有或资源冲突,则报错退出,否则继续;
 - d. 根据具体仪器功能接口配置文件调用该仪器实现 measure-DC-volt 功能的接口函数;
 - e. 具体仪器的功能接口函数调用该仪器的厂家驱动器函数;
 - f. 厂家驱动器函数调用相应的 VISA,IVI 库实现具体的仪器控制功能。

由以上具体实现过程可知,在功能接口函数内部封装了测试程序底层复杂、繁琐的基于驱动程序的仪器控制过程。功能接口函数与测试任务和测试资源的接口信息以配置文件的形式存储起来,执行时映射成具体的配置信息。因此,软件开发需要进行配置,包括 3 个步骤:仪器通道索引配置、UUT 节点和仪器通道映射以及具体仪器功能函数和测试功能接口函数的匹配。

1.2.2 配置工具和配置文件设计

1.2.2.1 仪器通道索引配置工具

仪器通道索引配置工具根据仪器通道数为仪器的每个通道分配 1 个整数索引,配置结果生成 1 个 .h 文件,格式如下所示:

```
//测试资源描述
//AD
#define hp1413 0
#define hp1413_MAX 63
//DA
#define hp1418 100
#define hp1418_MAX 115
//资源通道描述
#define AD0 0
#define AD1 1
#define AD2 2
#define AD3 3
.....
#define DA0 100
#define DA1 101
#define DA2 102
#define DA3 103
.....
```

测试功能接口函数执行时,参数 UUT-pin 被最终映射成具体仪器的通道索引值。当硬件资源改变后,只需要使用该工具对各个仪器进行配置,重新生成 .h 文件。

注意在设计配置工具生成资源通道整数索引时,不要出现索引值重合的现象,以免造成功能接口函数内部执行代码映射多重信息,而无法执行的错误后果。

1.2.2.2 测试节点映射工具

测试节点映射工具将测试节点与具体的物理仪器通道连接。使用该工具时,通过输入测试节点和具体的物理仪器通道信息及被测组合名称,将在指定的路径下生成一个以组合名称命名的 .h 文件,格式如下所示:

```
#define Test_1 AD0
#define Test_2 AD1
#define Test_3 AD2
#define Test_4 AD3
.....
#define Stim_0 DA0
#define Stim_1 DA1
#define Stim_2 DA2
#define Stim_3 DA3
.....
```

在针对不同 UUT 写测试程序时,只要使用该工具并根据具体的硬件连接关系进行相应的配置,生成 .h 文件并包含进工程即可。测试程序执行时会映射成相应的仪器通道标识符,并根据前面的仪器通道索引配置文件进一步映射成仪器通道整数索引,最终实现了 UUT 节点与仪器通道的一一对应关系。

1.2.2.3 仪器功能接口函数匹配工具

仪器功能接口函数匹配工具建立具体仪器功能函数与测试功能接口函数的映射关系。即将具体仪器实现某种测试功能的函数名称记录在采用 Windows 标准 INI 形式的配置文件中。

将所有用到的仪器驱动放到指定的路径下,该工具能够自动读取选择仪器的动态链接库,并将其中的导出函数列在“仪器对应函数名称”的下拉列表控件中,这样只要在封装仪器驱动的时候使导出函数名称和功能对应,则配置过程将非常简单。以 Tek 公司 GPIB 模块 tktds754 为例,配置文件形式如下所示:

```
[tktds754]
measure_DC_voltage = tktds754_dc_volt
measure_AC_voltage = tktds754_ac_volt
measure_AC_current = tktds754_ac_current
measure_DC_current = tktds754_dc_current
measure_res = tktds754_res
measure_period = tktds754_period
measure_frequence = tktds754_frequence
measure_rise_time = tktds754_rise_time
.....
```

基于以上映射关系,功能接口函数执行时,根据第 1 个通道参数判断出被测节点与哪个具体仪器相

连后,就可以通过读取 INI 文件得到要调用的具体仪器功能函数名称并进行调用,实现测试或激励功能。

1.3 模型特点

基于配置文件的虚拟仪器测试软件模型主要有 3 个特点。

a. 简化了测试软件开发。基于配置文件的虚拟仪器软件模型是面向 UUT 节点的“测试任务”的,配置工具有效地简化了软件开发过程。它把原先很多需要写代码才能实现的管理、设定任务转变成图形化界面点击、配置文件填写等简易操作,用户可以通过友好的接口实现软、硬件资源的集成。比如设计某个测试任务,可先用配置工具指定仪器的测试通道等相关参数项,然后命名此次测试任务,并保存在配置文件中。编程时,只要根据具体的测试任务需求调用相关接口函数并输入 UUT-pin 等参数,程序将自动在配置文件中搜索相关信息,映射成功后就可进行相应的功能操作,可以实现“测试任务”的现场组态。

b. 实现了代码重用。基于配置文件的虚拟仪器软件模型把整个仪器控制过程定义成测控服务层为用户应用程序和底层硬件的集成提供服务。测控服务层把仪器设备的通用功能进行封装,以软件模块,即功能接口函数库的形式提供给用户调用,这样可以有效地实现资源共享和代码重用,实现了多模块、多任务的无缝集成。更换仪器后,只要更改功能接口函数内部的实现代码并进行重新配置,而测试程序完全不用更改。

c. 方便了系统管理。配置工具把测试资源及其通道信息存储在配置文件中,它们为应用程序提供接口映射信息。通过查看相应的配置文件,用户可清楚地了解测试资源配置、测试通道映射、功能函数匹配、测试任务与 UUT 关联等信息,方便了系统管理。

2 虚拟仪器测试软件开发

如图 3 所示,基于配置文件的虚拟仪器软件包括测试程序、功能接口函数库和配置文件三部分。

图 3 中测试程序是具体“测试任务”的流程化实现,它由装备工程师根据 UUT 的测试需求调用功能

接口函数编写;功能接口函数库是测试资源面向“测试任务”的标准化接口,它由软件工程师根据具体的仪器设备和功能控制需求编写。配置文件存储了相应的软、硬件映射信息,由系统软、硬件设计人员使用配置工具进行配置。综上,基于配置文件的虚拟仪器软件设计实现了装备工程师、软件工程师和硬件开发人员的良好分工,简化了系统集成。

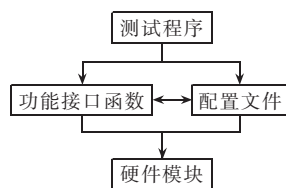


图 3 基于配置文件的测试软件结构

Fig.3 Architecture of test software based on configuration file

3 软件设计方案的改进

通过具体工程实践发现,基于配置文件的虚拟仪器软件设计方案^[6,7]在以下方面还需要进一步改进:

a. 模型中的功能接口函数是采用 C 语言开发的,跨平台、跨语言能力有限;

b. 功能接口函数种类和数量受平台测试功能和具体的测试任务需求影响,需要不断的扩展和更新。

3.1 基于 COM 技术实现功能接口组件化

COM 技术是 Microsoft 公司发布的软件设计规范,它提供了一种开发软件组件的方法,这些软件组件内部封装了技术实现细节,对外公布标准调用接口,提供某项服务。COM 接口是与语言、进程、平台无关的,经扩展,还可以跨机器边界访问,应用非常方便。为此,采用 COM 技术封装功能接口,实现功能接口的组件化,将进一步实现测控服务层的跨平台、跨语言移植支持。

3.2 面向测试和激励的功能接口

无论面向 UUT“测试任务”,还是面向“测试资源”,测试软件只包括测试和激励两种情况。为此,可以把功能接口进一步封装,而通过在接口函数中设定功能类型参数区分具体的功能种类。当功能接口种类和数量需要变更或增减时,只需更改接口内部实现就可实现。这样将进一步实现测试功能接口的标准化,更有利于面向“测试任务”的虚拟仪器软件设计。

参考文献:

- [1] 刘君华. 现代检测技术与测试系统设计[M]. 西安:西安交通大学出版社,1999.
- LIU Jun-hua. Technology and system design of modern measurement and test[M]. Xi'an:Xi'an Jiaotong University Press,1999.
- [2] 何 英, 孟 晨. 基于模糊神经网络的智能化虚拟仪器

[J]. 电力自动化设备,2002,22(5):26-28.

HE Ying,MENG Chen. The design of virtual instrument based on fuzzy neural network [J]. **Electric Power Automation Equipment**,2002,22(5):26-28.

- [3] 赵丽平,李群湛,陈小川. 基于虚拟仪器的接触网故障测距装置[J]. 电力自动化设备,2003,23(4):53-55.

ZHAO Li-ping,LI Qun-zhan,CHEN Xiao - chuan. Virtual instrument-based feeder locator [J]. **Electric Power Automation Equipment**,2003,23(4):53-55.

- [4] 张承学,张翌晖. 虚拟仪器中基于 197 的智能卡研制[J]. 电力自动化设备,2002,22(7):44-46.

ZHANG Cheng-xue,ZHANG Yi-hui. Development of MAX197-based intelligent card for virtual instruments[J]. **Electric Power Automation Equipment**,2002,22(7):44-46.

- [5] 杨锁昌,孟 晨,方 新. 利用引擎技术设计复杂装备检测系统[J]. 测控技术,2002,21(1):30-33.

YANG Suo-chang,MENG Chen,FANG Xin. Using engine technology to design test system for complex weapons[J]. **Measurement & Control Technology**,2002,21(1):30-33.

- [6] 刘金宁,孟 晨,方 新,等. 基于信号接口的 IVI 驱动器设计标准——IVI-Signal Interface 及应用 [J]. 测控技术,2004,23(4):68-70,73.

LIU Jin-ning,MENG Chen,FANG Xin,*et al.* IVI driver design standard based signal interface—IVI-Signal Interface and its application [J]. **Measurement & Control Technology**,2004,23(4):68-70,73.

- [7] 罗 锦,孟 晨,苏振中. 一种基于数据库的 VXI 总线测试软件设计[J]. 计算机测量与控制,2002,10(9):606-608.

LUO Jin,MENG Chen,SU Zhen-zhong. A design for the VXI bus test software based on database [J]. **Computer Measurement & Control**,2002,10(9):606-608.

(责任编辑:汪仪珍)

作者简介:

刘金宁(1979-),男,河北衡水人,博士研究生,主要从事装备自动检测与故障诊断方面研究(E-mail:fei2002909@eyou.com)。

Design and implementation of virtual instrument test software based on configuration file

LIU Jin-ning¹,MENG Chen¹,CUI Shao-hui¹,CHEN De-xiang²

(1.Ordnance Engineering College,Shijiazhuang 050003,China;

2.Missile Station of Nanjing Military Area,Nanjing 210028,China)

Abstract: Problems in software and hardware integration of virtual instrument test system are pointed out,such as poor universality,non-test-task-oriented,complex test platform and so on. A model of virtual instrument test software based on configuration file is designed. Its architecture and characteristics are analyzed and its implementation process is discussed. The virtual instrument software design based on the model is presented. The test software presented realizes the configurable software architecture,the reuse of software code and the interchange of hardware modules,which simplifies the system integration.

Key words: virtual instruments; test software; configuration file; function interface