

# 标准接口仪器测试应用软件辅助开发工具\*

张世琨<sup>1</sup> 王立福<sup>1</sup> 冯慧<sup>2</sup> 杨芙清<sup>1</sup>

<sup>1</sup>(北京大学计算机科学技术系 北京 100871)

<sup>2</sup>(航天工业总公司无线电计量测试研究所 北京 100854)

E-mail: zsk@cs.pku.edu.cn

**摘要** 领域工程是目前软件工程领域的一个重要研究方向,通过对基于特定领域的核心资产库的复用,可以提高开发效率和质量,降低维护成本.该文通过对基于标准接口仪器测试软件的领域分析,提出了该领域软件的体系结构,并开发了相应的支持工具.文章最后是使用该工具的体会和进一步的工作.

**关键词** 特定领域软件体系结构,仪器测试应用软件,虚拟仪器.

**中图法分类号** TP311

经过几十年的软件研究和开发实践,今天,人们开发的应用系统很少与过去的系统没有联系或相似之处,特别是属于同一应用领域中的系统,它们具有相似的软件需求,彼此复用的可能性更大.这是领域工程研究的出发点之一.

领域工程对领域中的系统进行分析,识别这些应用的共性和特性,形成领域模型,并设计面向特定领域的软件体系结构(domain-specific software architecture,简称 DSSA).当开发同一领域中的新应用系统时,以领域模型和面向特定领域的软件体系结构为框架,确定系统需求,并进行体系结构设计,以此为基础组织构件进行系统开发.<sup>[1~3]</sup>

在国家“九五”重点科技攻关项目《软件工程环境(青鸟 CASE)工业化生产技术及系统的研究开发》中,我们研制开发了“标准接口仪器自动测试软件开发平台”.通过对基于标准接口的仪器自动测试软件的领域分析,我们提取该领域应用系统的典型特征,建立了领域需求模型,提炼出软件体系结构,并开发了一个基于 DSSA 的仪器测试软件辅助开发工具(computer-aided instrument testing tool,简称 CAIT 工具).它可以辅助用户自动生成仪器测试应用软件的框架.用户可以使用已经存在的构件或开发新的构件,组装到生成的框架中,完成测试应用软件开发.

## 1 仪器测试软件系统体系结构

### 1.1 LabView VIs 的结构

CAIT 工具基于 National Instrument 公司的 LabView 平台,LabView 采用可视化方式开发仪器控制应用程序,支持虚拟仪器(virtual instrument,简称 VI)的概念.每个 VI 由面板(panel)、框图(diagram)以及连接器(connector)等成分组成.面板是 VIs 的人机交互界面,模拟实际仪器的控制面板;框图是由可视化的图元构成的程序,具体实现 VIs 的功能;连接器是 VIs 的外部数据接口,它包含 VIs 的输入、输出信息,如 I/O 参数的个数、类型等,类似于传统编程语言中函数或过程的参数列表.图 1 给出了计数器 VI 的图标和连接器,图标左侧是 VI 的

\* 本文研究得到国家“九五”重点科技攻关项目基金(No. 98-780-01)资助.作者张世琨,1969年生,博士生,讲师,主要研究领域为软件工程,软件体系结构.王立福,1945年生,教授,主要研究领域为软件工程,软件测试、信息安全.冯慧,女,1953年生,高级工程师,主要研究领域为计算机辅助自动测试.杨芙清,女,1932年生,教授,博士生导师,中国科学院院士,主要研究领域为软件,软件工程,软件工程环境.

本文通讯联系人:张世琨,北京 100871,北京大学计算机科学技术系

本文 1999-06-25 收到原稿,1999-09-07 收到修改稿

输入部分,右侧是输出部分<sup>[4]</sup>.

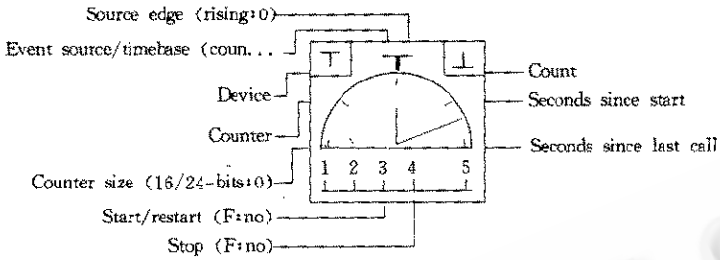


Fig. 1 Input/Output of a counter virtual instrument  
图1 一个计数器虚拟仪器的输入/输出

在 LabView 中,每个应用程序是一个 VI,它由一组协同工作的子 VIs 构成,这些子 VIs 可以进而由更小的 VIs 复合而成,形成一个层次系统. VIs 是 CAIT 工具生成的应用软件的基本构件.

### 1.2 标准接口仪器测试应用软件 DSSA

通过对基于标准接口的仪器自动测试软件的领域分析,一个测试应用软件的一般工作流程如下:仪器初始化、测试项目设置、数据采集、计算监视、数据浏览、原始数据入库和测试证书生成.图 2 给出了仪器测试应用软件的体系结构(即 DSSA).其中,数据库存放的是有关计量检定业务的各种数据,包括标准仪器和被检仪器的登记信息、仪器类型、各种类型对应的测试项目和测试指标、测试规程、原始测试数据和证书信息等.

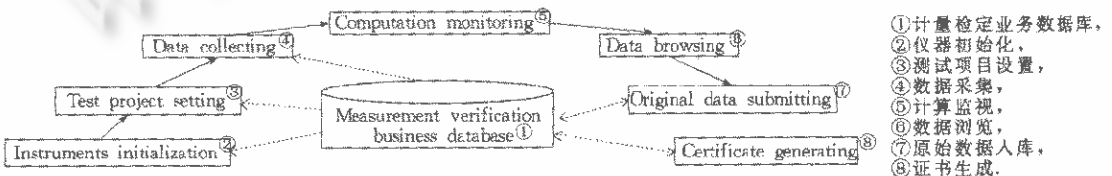


Fig. 2 Architecture of the instrument testing applications  
图2 仪器测试应用软件的体系结构

- ① 计量检定业务数据库,
- ② 仪器初始化,
- ③ 测试项目设置,
- ④ 数据采集,
- ⑤ 计算监视,
- ⑥ 数据浏览,
- ⑦ 原始数据入库,
- ⑧ 证书生成.

该 DSSA 体现了混合式的体系结构风格:数据库、管道-过滤器和层次化风格<sup>[5]</sup>,其中每个构件的功能如下:

(1) 仪器初始化:一个自动测试系统中的测试仪器和被测仪器通过标准接口(如 IEEE488, VXI, PXI 等)与计算机相连.在仪器使用之前,必须先进行初始化;

(2) 测试项目设置:一个被测仪器可能具有多种功能,对应着多个不同的测试项目,但每次检测不一定需要测试所有的项目,因此,用户可以选择相应的测试项目;

(3) 数据采集:根据测试项目对应的各项测试指标,向被检仪器发送控制信号,采集仪器产生的各种原始测试数据;

(4) 计算监视:为改善测试过程中测试系统与测试人员的交互,可以在测试过程中实时地监视当前项目的测试情况,并对数据进行计算、分析,以方便测试人员理解的方式将测试结果展现出来,如趋势图、变化曲线等;

(5) 数据浏览:当一个项目测试完成以后,在测试结果保存在数据库之前,测试人员有机会再检查和确认测试结果;

(6) 原始数据入库:计量检定行业的有关规范和标准要求原始测试数据必须长期保存,以备检索和查询.该构件将测试过程中产生的原始数据,按照规定的格式存放到数据库中;

(7) 证书生成:根据数据库中存放的被测仪器的登记信息和原始测试数据,按照规定的格式自动完成测试证书的生成.

系统中的各个构件都是标准的 VI,具体的测试应用系统是通过对各抽象构件进行实例化而得到的.实例化的方法是直接复用领域资源库中相应的构件,或者根据需要开发新的构件.然后,在统一的体系结构框架下,通过 VIs 的接口进行集成.从而形成针对标准接口仪器测试应用软件家族的生产线<sup>[6]</sup>.

## 2 辅助开发工具

基于以上的体系结构和可复用构件,我们开发了支持开发测试应用系统的辅助生成工具,其结构如图3所示,在该工具的支持下,测试应用系统的开发分为两个阶段:系统框架生成和集成组装,计量检定业务数据库和可复用资源库分别对以上两个阶段进行支持。

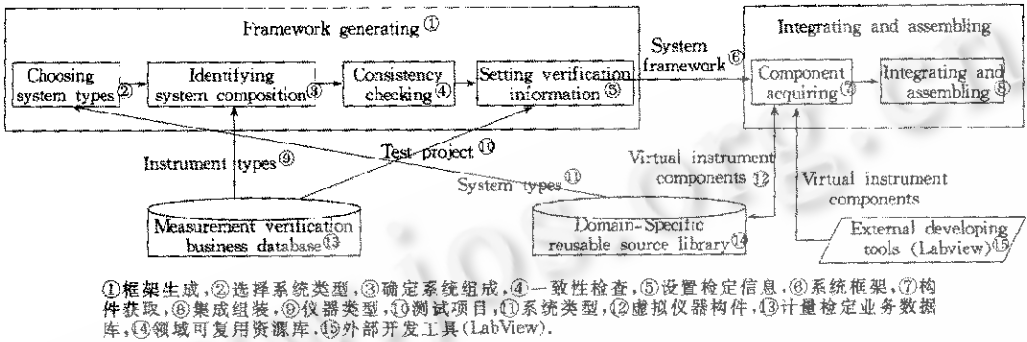


Fig. 3 Structure of the CAIT toolset

图3 CAIT工具的结构示意图

系统框架生成阶段首先根据测试系统的组成结构选择系统类型,例如一源、多源、一表、多表等,不同的系统类型对应于不同的处理模式;根据系统类型和检定规程,设置组成一个测试系统的标准仪器和被检仪器,并为其分配设备地址;检查标准仪器和被检仪器的地址有无冲突;选择被检仪器对应的测试项目、功能性检查项目和状态设置项目。

集成组装阶段的第1步是获取构件,主要是对应于上面的体系结构中的各个构件,例如仪器初始化、数据采集、计算监视等构件。构件获取可以直接复用资源库中相应的构件,或在外部开发工具的支持下开发新的构件,新开发的构件也可以加入到可复用资源库中,以备后来的系统复用。最后,通过框架集成各个构件,形成一个完整的应用系统。

## 3 体会和展望

目前,CAIT工具已开发完成,并投入实际应用。与传统的开发方法相比,它具有开发速度快、效率高、结构及界面统一等特点,大大降低了开发人员的劳动强度。进一步的工作包括完善系统体系结构中的构件接口;丰富可复用资源库的内容,提供更多的可以直接复用的构件。

我们的体会是,通过提炼面向特定应用领域的需求模型,并构造该领域的DSSA,以指导领域构件的开发和集成组装,进而建立面向该领域的产品生产线,这是软件工程化开发和工业化生产的必由之路。

## 参考文献

- 1 Len Bass, Clements P, Kazman R. Software Architecture in Practice. Reading, MA; Addison-Wesley Publishing Company, 1997
- 2 LTC Erik Mettala, Graham M. The domain-specific software architecture program. Special Report CMU/SEI-92-SR-009. Software Engineering Institute, Carnegie Mellon University, 1992
- 3 Clements P, Weiderman N. Report on the second international workshop on development and evolution of software architectures for product families. Special Report CMU/SEI-98-SR-003. Software Engineering Institute, Carnegie Mellon University, 1998
- 4 LabVIEW User Manual for Windows. National Instruments Corporation, USA, 1994
- 5 Shaw M, Gerlan D. Software Architecture: Perspectives on an Emerging Discipline. Englewood Cliffs, NJ; Prentice Hall, Inc., 1996

- 6 Bäumer D, Gryczan G, Knoll R *et al.* Framework development for large systems. *Communications of the ACM*, 1997,40(10):52~59

## Computer-Aided Developing Tool for Standard Interface Instrument Testing Applications

ZHANG Shi-kun<sup>1</sup> WANG Li-fu<sup>2</sup> FENG Hui<sup>2</sup> YANG Fu-qing<sup>1</sup>

<sup>1</sup>(*Department of Computer Science and Technology Beijing University Beijing 100871*)

<sup>2</sup>(*Institute of Radio Metrology and Measurement of China Aerospace Corporation Beijing 100854*)

**Abstract** Domain engineering is one of the important research directions in software engineering field. With domain-specific core assets, software reuse can promote development efficiency and quality, reduce maintenance cost. Based on the domain analysis of standard interface instrument testing software, the authors propose DSSA (domain-specific software architecture) for this domain, and develop supporting tools in this paper. The experience gained and the further work are provided as the conclusion.

**Key words** DSSA (domain-specific software architecture), instrument testing application, virtual instrument.