

浅谈面向对象的ATE设计

上海聚星仪器有限公司

摘要: 面向对象是现代软件工程的设计方法。近代科学家和工程师将其拓展到了系统分析和设计。这个方法使得软件系统设计更加符合实际应用,更加灵活,易于管理,易于升级维护。本文讲述了什么是面向对象的系统设计,并将其应用到一种射频、音频自动测试ATE设计。

1 前言

软件定义仪器是现代计算机辅助测量的主流方法。不论是台式仪器还是模块化仪器,不论是标准仪器还是定制仪器,都倾向于采用较通用的硬件组件或模块,通过软件控制完成信号调理采集,并用软件进行分析显示和数据存储。这样大大提高了硬件设计的重利用率,使得新仪器更快推出,新需求更快得到满足。当计算机硬件技术被融入到仪器,计算机科学理论和方法也就成为了仪器科学的“助推火箭”。

这个助推火箭解决的最大问题是系统灵活性适应客户需求更改,系统模块化适应开发人员合作和流动。仪器学科作为电子科学技术和计算机科学技术的边缘科学技术,必须充分吸收和发扬相关科学的最新成果,才能创造出可持续发展的道路。

2 面相对象的设计方法

2.1 历史

“对象”英语“Object”在计算机科学里面1960年出现在MIT的LISP编程语言^[1]。但当时Object仅仅是指编程的原子符号。最早的面向对象的计算机语言,大家认为是挪威计算中心Kristen Nygaard和Ole-Johan Dahl在1960年代后期发布的离散事件仿真计算机语言Simula 67^[2-3]。在Simula 软件里面引入了类、对象、属性等面向对象的概念。在1970年代施乐Palo Alto实验室(Xerox PARC)的Alan Kay主持设计了SmallTalk语言。这是一个面向教育的,较完整的面向对象编程语言。后来的OOP语言大多受到它的影响。有趣的是Alan Kay在1972年设计了一个给儿童使用的平板电脑“Dynabook”。可惜到2008年他才拿出模型来秀。说来PARC也是一个神公司,看看乔布斯传,查查维基百

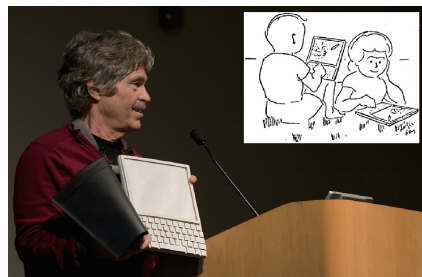


图1 Alan Kay和他的“Dynabook”
电脑模型, 2008年^[4-5]

科就会发现鼠标、视窗操作系统、个人电脑、网路服务器等发明都有它的份。作为硅谷邻居,乔布斯和盖茨都获益良多。

面向对象的系统设计起源于计算机语言,但是真正的系统开发方法要到1990年代才浮出水面。

1981年成立的Rational Software公司致力于提供现代软件工程工具。该公司的Grady Booch在1991出版了《面向对象的分析和设计》^[6]一书,成为经典。Booch和Rational Software另外2位工程师一起,吸收了各人创造的方

法和业界主流研发流程，创立了统一建模语言(unified modeling language)，提出了合理化统一流程(rational unified process, RUP)。2003年IBM以21亿美元收购了Rational Software公司。2005年ISO接纳UML成为ISO/IEC 19501标准。2008年Booch担任IBM软件工程首席科学家。

2.2 方法

面向对象的分析和设计也基本遵循项目工程或者软件工程的步骤，其主要工作阶段在于需求分析、概要设计、详细设计阶段。这些阶段里面主要的工作是通过建立操作员和系统交互的“用例”(Use Case)，定义系统工作方式方法。在面向对象分析设计理念里面，设计师按照目标系统自然的对象封装为软件类，使得类与类之间耦合度降低，获得扩展的灵活性和修改的封闭性。所以，设计师在软件总体设计时不企图定义一个大而全的架构，反而倾向于掌握所有客户需要的用例，并设计软件模型来满足这些用例。

而设计软件模型的过程，又叫对象建模。过去面向过程的软件将软件抽象成算法加数据，两者分开来考虑，导致客户用例和算法，算法和数据之间产生较大概念隔阂需要设计弥补。而软件对象类将方法和属性，或者说动作和数据封装在各个对象里面，成为高内聚度的整体。这样的面

向对象技术使得软件更可靠，更灵活，更可重用。

面向对象的分析和建模的产出是一系列文档。计算机科学界常用UML表述。UML是综合了各种软件工程方法的图形化语言，有非常多的图标模式。作为产业应用，可以根据实际应用规模，开发人员基础，挑选适当的子集来帮助模型可视化。UML当中用例图、类图、序列图、状态图是经常有用的。

3 面向对象的ATE设计

假设一个简单的ATE，测试某一

种被测件UUT(unit under test)。操作员启动登录，系统准备，进入测试循环，每循环一次测试一个UUT，直到测试完成，退出测试，关闭测试流程，操作员登出。用状态图描述如图2所示。实际上，有很多状态在测试内部。但是测试流程就那么简单，按顺序执行，遇到测试结果超限可以选择重测，若可重测，则重测若干次通过一次即可通过。无论通不通过都要记录结果。通过则进行下一测试步骤，不通过可选择执行下一步骤或结束本次UUT测试。这样的状态运行可

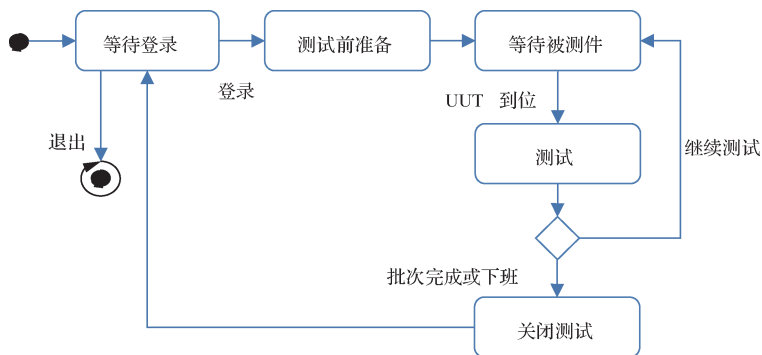


图2 简单ATE状态图

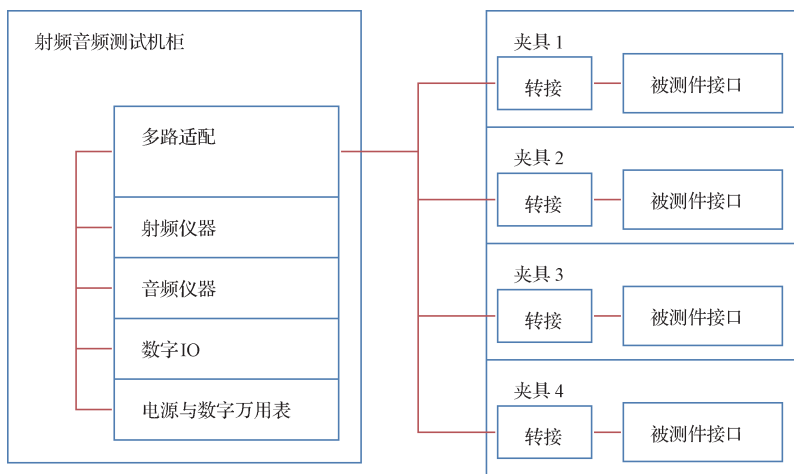


图3 范例ATE原理

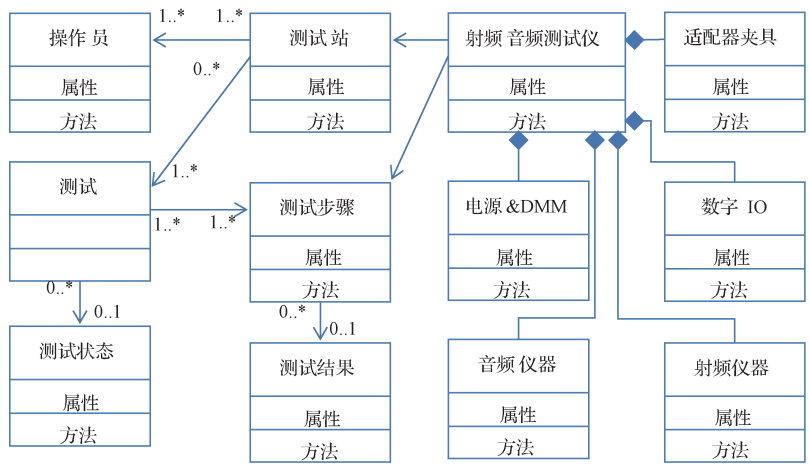


图4 简化的ATE类图

以通过定义相应的类，以及对象实例调用来实施。如图4所示为一个简化的类图。类图是描述类和类之间关系的图形文档。简单地，类可以用一个3行单列表格表示，其中第一行是类名称，第二行是类包含的属性，第三行是类包含的方法。在简化类图里面暂时没有写，而在如图5所示中描述了一个较完整的测试步骤类。这个类包含设置一个测试步骤实例的子步骤

(命名为“set.....”)、获取状态(命名为“get.....”)和运行步骤的方法，也包含可读写的属性。这个例子里面属性包括是否可以失败重测，最多重测几次。在图4里面还通过箭头和数字表述了类的关系。在UML里面不同端点形状的连线表述不同的关系，主要包括接口、依赖、泛化、聚合、组合、关联、实现。在本图使用了单线箭头表述的关联关系，用实心棱形箭头表述组合关系。箭头类型意义如图6所示。

测试 步骤
失败可复测: boolean
最大复测次数: int
set 夹具动作() set 信号路由() set 信号激励() t set 测量() set 判定() set 步骤记录() get 夹具状态(): 夹具状态 get 测试结果(): 测试结果 run 测试步骤(): Int

图5 测试步骤类定义

本文借用《国外电子测量技术》2015年第12期刊登的《非标系统标准化制造 - ATE的成功之道》中讲述

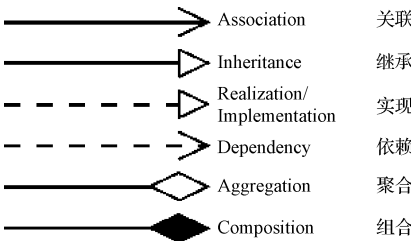


图6 类图箭头表述类的关系

的设计案例。这个ATE的原理如图3所示，包括各种仪器、适配器和夹具。ATE类图设计时将用户要求和硬件设计作为输入条件，规划设计各个类和类之间的关系。在本案例，用户要求以系统状态图和硬件原理框图为输入条件，设计了类图，如图4所示。

类图设计完就可以开始软件代码设计。由于采用面向对象的分析设计方法，自然要选用面向对象的编程语言。考虑到ATE在Windows操作系统运行，需要通过各种总线接口、通信接口和硬件打交道，选用微软Visual C#。Visual C#是面向对象基于.NET架构的高级编程语言。而.NET把Windows下面各种接口、内存管理封装起来成为强大易用的统一界面。根据2016年9月TIOBE指数(TIOBE Index)统计报告，C#已经成为受欢迎排名第4的编程语言。在.NET和MONO的支持下，C#可以方便地和其他语言接口，甚至移植到其他操作系统。尤其是其对Windows对象天然的支持并开放给用户程序调用，使得编程特别方便，功能特别丰富强大。由于封装妥善和面向对象架构的利用，使得ATE测试流程可以在一个C#程序里面完成，而不再需要中间添加一个测试流程管理软件，从而进一步提高了可靠性，消减了过度包装。

4 方法总结

优秀的定制化ATE系统要满足多

种条件。主要条件包括满足客户需求,软件可靠灵活,软件架构和模块重利用率高,文档可持续维护。软件定义的仪器在电子电气方面提供了良好的硬件基础,而吸取计算机科学1990年代以来成熟的面向对象方法可以在软件方面提供另一半基础。

面向对象仪器设计除了软件工程的OOAD方法之外还要考虑与硬件设计的配合。硬件采用模块化界面化设计后,也天然具备了高内聚的低耦合度的特性,所以主要的将硬件实体抽象成类基本上无大错。

本文以一个简单的ATE设计为例,通过状态图和电气原理图分析,综合考虑用户需求和系统硬件架构,设计出软件的类和类关系,并用类图

描述。

总之,电子工程模块化设计加上软件工程面向对象设计成为面向对象的仪器仪表设计方法。电子电气工程师理智地取舍软件工程诸如UML等工具,使得仪器设计高效、可靠、灵活、可重利用。

参考文献

- [1] MCCARTHY J, BRAYTON R. LISP I Programmers Manual[S]. Boston, Massachusetts: Artificial Intelligence Group, M.I.T. Computation Center and Research Laboratory.
- [2] DAHL O J (2004). The birth of object orientation: the simula languages[J]. From Object-Orientation to Formal Methods, Essays in Memory of Ole-

Johan Dahl,2004:79-90.

- [3] HOLMEVIK J R. (1994). Compiling simula:a historical study of technological genesis[J]. IEEE Annals of the History of Computing, 1994,16(4), 25-37.
- [4] Wikipedia照片, By Marcin Wichary from San Francisco, U.S.A.-Alan Kay and the prototype of Dynabook.
- [5] http://www.vpri.org/pdf/hc_pers_comp_for_children.pdf[S].2016-10-06.
- [6] Butch B G. Object-oriented analysis and design with applications[S]. Redwood City the Benjamin/cummings Publ. comp.[c Ed(3), 2012: 1275-1279.
- [7] BOOCH G. Object-oriented analysis and design with applications[M]. 3rd Edition. Addison Wesley Longman Publishing Co. Inc,2007.