

利用面向对象的软件工程定制仪器

邵 晖

上海聚星仪器有限公司

摘 要: 本文通过一个车载数据记录仪案例,讲解如何运用面向对象的软件工程方法定制仪器。并且讲解了该方法通过基于对象的建模和流程管理,建立较完善文档、积累技术价值、降低人员依赖、减小软件缺陷损失。本文是利用电子工程结合计算机科学的方法在仪器仪表研制方面的创新探讨。

1 引 言

仪器科学技术是电子工程和计算机科学的交叉科学。日常科研工作中,将计算机装到仪器里或者仪器装到计算机里屡见不鲜。然而计算机科学的最新方法,尤其是软件工程的最新方法,还很值得仪器仪表业借鉴。

2 软件工程的任务

对于现代商业,最大的两个难题是项目成功和长期技术积累。项目成功是指在人力经费符合预算的情况下,按时良好地达到客户要求。这个美好的设想往往被现实当中客户需求变更而打破。长期技术积累是指完成一个新项目就成熟一个产品,而且老项目的设计和程序可以在类似项目里面多次重用。这样每一个新项目的研发费用就成为公司无形资产的投资积累。这个理想又被人员流动、技术更新而严重损耗。

除了用“人生就是修行”来自我安慰之外,可以看到计算机软件行业已经通过技术革命积极地摆脱如此的

煎熬。IT行业3个月一个新型号半年一个新产品已不稀奇,改变已是常态。另一方面IT行业也适应了平均2~3年换一个工作的人员流动速度。

面向对象的软件工程,就是用来解决这些问题的。

3 面向对象和结构化设计的区别

在仪器学科的计算机基础课里面可能听过,算法加数据结构就是软件。这个是传统结构化系统设计(structured system design, SSD)的概念。这种方法从计算机架构出发,寻求满足客户需求的解决方案。当面对直接的采集-分析-显示测量应用时,这个方法简单易用高效。但是当面对复杂逻辑、多机协同、人机互动的应用,这种方法设计出来的软件架构往往非常复杂,而且软件模块和仪器电子学模块产生多对多的复杂逻辑关系。这使得客户合理的需求变动会牵连大量软件模块及其互动关系的设计变更。从另一个角度描述,就是

结构化系统设计的低耦合度模块关系投射到客户应用域模型产生了混乱,而客户应用域清晰的要求投射到解决方案域时,各个软件模块是广泛耦合的。

举一个例子,本公司接到一个客户直升机遥感数据采集吊舱设计的项目。这个项目要采集遥感传感器接收到的地球物理信号、吊舱和传感器状态信息、GPS定位信息,也要接受直升机里面人工操作。没有经验的软件工程师设计了一个“巧妙”的数据流,把各个数据源的数据传递到统一的数据存储记录服务模块。但是到实际飞行的野外现场,发现GPS数据会发生不可靠,而GPS数据中断导致了整个数据服务中断。除了软件工程师额外花时间重写软件和调试之外,飞行测试效果不理想,带来更大损失。这个就是计算机数据链路的包装和实际现场对象数据流不一致,使得现场的单数据故障导致整个数据链路死锁故障。

更深刻的思考,是这个结构化成

序员过度沉迷于算法的“巧妙”，而使得解决方案域的软件模型和应用域的物理对象模块分割产生较大背离。单独从软件模型看是模块化的，但是当单一的故障从应用域对象传递到解决方案域的软件时，映射到了很多模块和复杂的耦合关系，导致问题恶化和修正难度加大。

从另一个方面看，结构化设计将处理算法和数据分开考虑，而处理模块是软件低耦合度的优化对象，使得数据流带来的模块关联耦合容易被忽略。甚至于在功能处理之外的易用性和客户体验也不被强调。这样导致软件模块上看起来设计合理耦合度低，

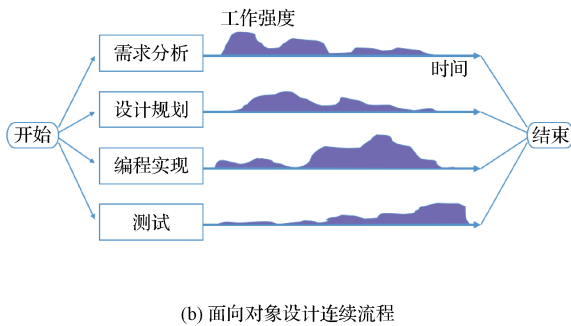
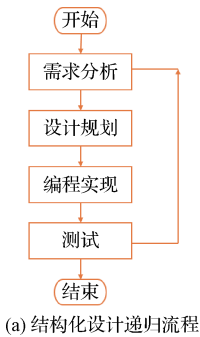


图1 面向对象软件工程的连续流程

但是面对应用修改和新发生状况，往往牵连一大批模块。

4 面向对象设计的基本方法和UML

首先是项目流程方面，面向对象的设计预计到客户需求会不断完善，复杂的项目在较长的生命周期里也会出现研发人员更换，研发产品交付后会有另外一批工程师来接手客户进行服务支持。基于这样的假设条件，面向对象的设计流程就是持续演进的流程。如图1所示，项目开始后，需求分析、设计规划、编程实现和测试会同时进行。只是各个工作在不同时间段的比重不同。

在系统设计阶段，需要做5件事^[2]：
 1) 理解和定义系统上下文和外部交互、
 2) 设计(选择)系统架构、
 3) 确认系统包含的主要对象、
 4) 建立设计模型、
 5) 定义接口和界面。这一系列的工作输出应该是用户看得懂的文档。而这些文档就是通过交流建立起来的各种模型说明。

研发工作由客户交流和模型建立开始。客户交流对于项目顺利至关重

要。即使我们对客户需求改变做好了准备，也要知道什么时候怎样去和客户交流。研发团队针对客户需求建立应用域的模型。而这个模型就可以用统一建模语言(UML)来描述。统一建模语言是1990年代发展成熟的，并被ISO接纳为国际标准^[5]。由于工程资源投入和活跃程度，商业界更多偏向用OMG(object management group)标准协会的标准^[4]。对于应用域模型，

可以用“用例”来描述。而用例可以用UML的用例图(Use Case Diagram)来表述。当然，笔者体会，电子工程常用的操作流程图或者作业指南文字，都可以成为用例的描述。用什么来描述，取决于讲得清楚、看得明白、符号通用。

这些符号对于面向对象的设计，需要讲清楚3种模型^[1]：功能模型(用例图)、对象模型(类图)、动态模型(交互图、状态图、活动图)。OMG标准提供了十几种框图帮助模型文档化^[4]。

如图2所示，给出了一个用例图范例。我们设计一个车载数据记录仪。该用例图表述了操作员要进行2种使用操作：数据记录和数据回放。还可以进一步细化需求，比如，将数据记录分作车载记录仪配置和车载记录仪启停；将数据回放细化成数据文件检索和数据浏览。无论怎样细化，这是一个一目了然的起点，让设计师和用户开始清晰完整的基于面向对象模型的沟通。

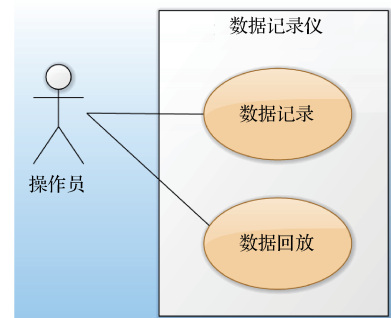


图2 数据记录仪用例图范例

接下来就是要建立解决方案的模型。仪器系统解决方案和硬件不可分离，所以电子工程的硬件设计相对

先行。当有了符合客户需求的硬件设计之后，就可以进行较详细的软件设计。硬件设计输出了硬件原理框图和所有硬件配置的使用手册，软件设计就以此为输入，建立面向对象的模型。

软件可以作为系统分解为子系统，而子系统还可以分解，直到模型足够简单，可以一目了然。而这些分解到最细的子系统，就由各种对象构成。软件对象有多种定义，实际上就是计算机软件设计对现实目标的抽象，这种抽象把目标数据和处理封装在一起成为属性和方法。有人称软件里面对信息产生反应的模块为对象。对象的抽象定义成为

类，类的实例就是对象。在一个软件子系统里面，对象的定义和关系描述就可以用“类图”。

如图3所示为车载数据记录仪的构成。该系统包括车载记录仪和监控终端。多个车载记录仪各自布设在动车牵引车厢，采集记录各种物理信号，并通过无线网络传送到监控终端。

如图4所示为非常概要的监控终端概要类图。其中每一个方框分成3行，就表示一个类。每个类框的第一行是类的名称。类框的第二行就是属性，亦即类内部的，对外可见的变量。类框的第三行就是类的方法，也就是类可以干什么。类框之间的连线描述了类之

间的关系。而类的划分往往接近硬件模块的分工，这样就使得应用域到解决方案域的映射比较直接、易懂、容易维护。

监控终端有2个接口界面，人机界面和远程监控车载记录仪界面。在这个类图设计里面，安排车载记录仪状态显示和车载记录仪控制两个类，负责人机界面。设计远程通信类负责和各个车载记录仪的网络接口界面。在监控终端子系统内部，设计了状态信息处理，用来处理车载记录仪的状态和波形数据，计算虚拟通道，频谱分析和其他特征值。另一方面设计了事件逻辑，控制在设定的时间间隔触发远程通信，获得车载记录仪信息，或者触发状态显示，调用状态信息处理显示状态和报警给操作员。这样的界面—处理—控制逻辑分配，也符合MVC(model-view-control)软件架构模型，使得模块内聚度提高，模块间耦合度降低。

类之间有动态和静态交互模型。可以选择交互图、状态图或活动图来描述。表示对象间交互的图叫序列图。如图5所示为数据记录仪3个基本对象的2种交互。一种是定时发生的状态汇报。车载记录仪定时状态发布给遥控监测终端(监控终端)，监控终端接收到状态发布后状态更新给操作员。另一种是操作员发起的指令操作。操作员对监控终端下达指令，终端将指令翻译为遥控指令下达给车载记录仪。车载记录仪将指令回复发

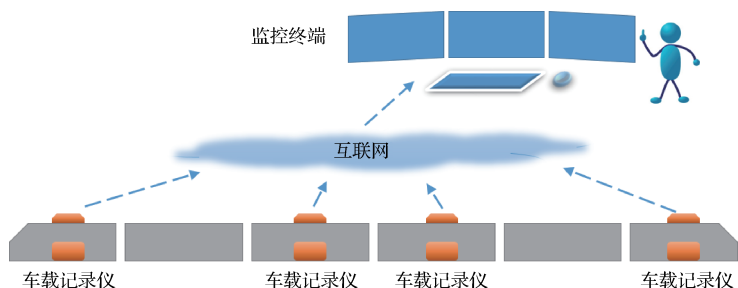


图3 车载数据记录仪系统原理

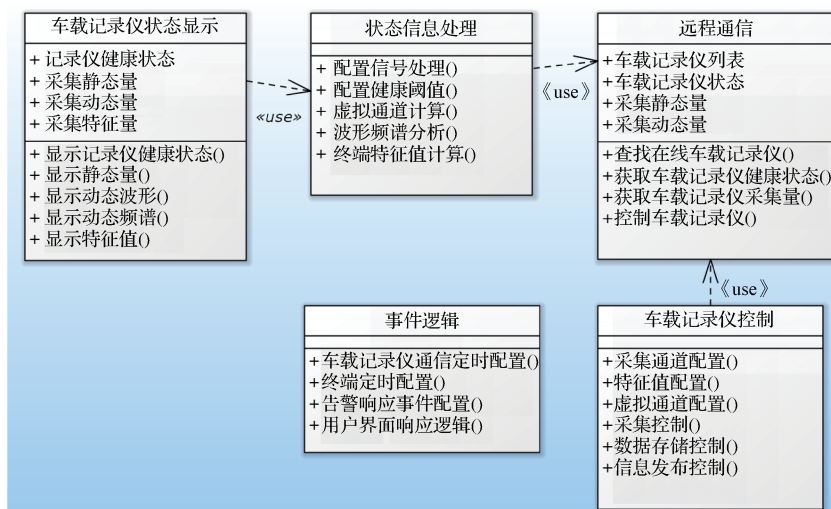


图4 监控终端概要类图

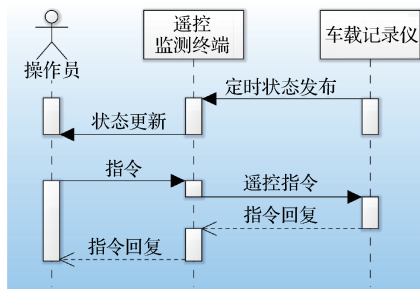


图5 数据记录仪系统序列示意

给监控终端，终端再翻译显示给操作人员。这个序列图只是较宏观的设计，在详细设计里面，还要将通信建立、鉴权、异常处理机制等细节完善。

5 测试设计

很多程序员问“我程序还没有写，怎么知道如何测试呢？”这是基本概念的错误，软件目的是满足客户应用需要，所以明确应用需求也就明确了顶层测试目的。将测试分成3个层次^[1]，系统测试、集成测试和单元测试。由于用模型设计软件，测试就是将软件产品和模型进行对比。系统测试在需求分析阶段策划，它通过正常使用场景和可能的异常操作验证软件和需求模型的一致性。集成测试在系统设计时策划，它测试软件子系统的集成和系统设计模型的一致性。单元测试在对象设计时策划，它测试软件单元和对象设计模型的一致性。

这样的计划安排，就是软件工程教科书里面说的，使用说明和验收大纲要在软件代码之前完成。在自律的软件开发流程管理下，项目风险得到

更好的控制。

另一方面，当有合理充分的软件文档，就可以聘请专职的测试工程师来进行测试。测试工程师可以根据应用模型和解决方案模型设计各种测试，在确保软件质量的同时降低研发成本。

如表1所示为美国国家标准和技术研究所的报告^[6]，它表述了软件缺

陷在发生的阶段更正代价最小，其后代价迅速呈指数放大。这个2002年的报告还指出软件缺陷给美国经济带来一年大约595亿美元的损失，其中大约1/3是软件开发缺陷造成。当有完整的模型文档，就可以在项目早期进行检验和测试，使得软件设计缺陷及早被发现和修正。

表1 不同阶段缺陷更正的代价

| 缺陷更正代价 | 缺陷发现时间 | | | | | |
|--------|--------------|-------------|-------------|------|-----|-----|
| | 需求分析 架构设计 | 编程/ 单元测试 | 集成/ 系统测试 | 客户评测 | 发布后 | |
| 缺陷引入 | 需求和架构 | 1x | 5x | 10x | 15x | 30x |
| 时间 | 编程 | — | 1x | 10x | 20x | 30x |
| | 集成 | — | — | 1x | 10x | 20x |

6 总结

面向对象的软件工程给定制仪器设计带来了经过实践验证的流程、方法和工具。通过使用面向对象的工程流程、系统分析和设计、统一建模语言等计算机科学的成果，可以更好地满足客户逐步成熟的需求，设计出更可靠更灵活的软件，积累更多可重利用的系统架构和模式。同时，齐全的标准文档和及早的测试，使得我们可以应对人员更换、降低测试和缺陷更正的成本。这种交叉学科相互借鉴的方法给测试测量系统定制带来了新的活力。

参考文献

[1] ALLEN H D, BERND B.面向对象软件工程[M].第三版.北京:清华大学出版社,2011.

[2] IAN S. 软件工程[M].程成,译.第9版.北京:机械工业出版社,2011.
 [3] 王菁,赵元庆.UML建模 设计与分析标准教程[M].北京:清华大学出版社,2013.
 [4] OMG Unified Modeling Language, version 2.5[EB/OL].(2016.11.10) <http://www.omg.org/spec/UML/2.5>.
 [5] ISO/IEC 19501:2005 Information technology—Unified Modeling Language (UML) Version 1.4.2[Z].
 [6] The Economic Impacts of Inadequate Infrastructure for Software Testing, Planning Report 02-3, NIST, 2002[Z].