

UML 2.0 述评^{*}

马浩海 邵维忠 麻志毅

(北京大学信息科学与技术学院 北京 100871)

摘要 在经过 4 年之久的修订过程之后,OMG 采纳了 UML 2.0。本文首先介绍了 UML 2.0 的修订背景和提案需求;然后分别介绍了 UML 2.0 规范的主体部分:基础结构和上层结构的修订特点,进而总结了 UML 2.0 基础结构的设计目标和体系结构,归纳了上层结构的设计目标、体系结构组织,着重于比较 UML 1.X 和 2.0 在模型构造能力上显著增强的部分;最后针对 UML 2.0 规范及对其存在的问题给出了一些评述,这些可以作为今后相关研究工作的框架。

关键词 UML 2.0, 基础结构, 上层结构

An Overview of UML 2.0

MA Hao-Hai SHAO Wei-Zhong MA Zhi-Yi

(School of IT Science and Technology, Beijing University, Beijing 100871)

Abstract After 4 years' revision process, UML 2.0 is adopted by OMG as a newer specification for the modeling language. First, this paper discusses why UML should be revised and what requirements are concluded in UML 2.0 RFPs. Then the paper reviews the two main parts of UML 2.0 specifications, i.e. Infrastructure and Superstructure. The design goals and architectural formalism of two parts are come up separately in the following section. This paper focuses on the contrast of modeling capabilities between UML 1.X and UML 2.0. Superstructure and several significant augments are illustrated with examples. Finally, some comments and critical thinking related to UML 2.0 are mentioned in the paper, which can sever as frameworks for following research.

Keywords UML 2.0, Infrastructure, Superstructure

1 前言

UML (Unified Modeling Language) 是 OMG (Object Management Group) 于 1997 年采纳的,用于软件系统的可视化、详述、构造和文档化的统一建模语言^[1]。发展至今,人们使用 UML 的目的已经从对基本的系统结构和行为的可视化描述,发展到利用 UML 实现程序代码的图形表示、系统行为的精确定义,以及模型的自动验证和执行^[2]。OMG 在 2001 年提出的模型驱动体系结构(Model Driven Architecture),甚至希望利用 UML 来代替编程语言,彻底对现在的软件开发方式产生革命性的变革^[3]。

虽然 UML 得到了广泛的应用,但是更普及的应用需求,使得人们不断发现它有一些缺陷和不足^[4,6,7]。概括起来说,主要体现在:体态臃肿和无理由的复杂性;部分模型构造物的语义不精确;对基于构件的软件开发(CBSD)支持不够;缺乏模型的编译、验证和执行能力;对元模型的定制能力有太多的限制从而使其在不同领域的应用受到制约;各种建模工具没有标准的实现规范,使得不同工具产生的模型缺乏交换能力;等等。所以,当 OMG 在 1999 年提出 UML 2.0 的修订信息需求(RFI)时,来自不同领域的 26 个响应者大多数建议应该对 UML 进行一次重大的修订^[4]。之后,OMG 于 2000 年初发布了 UML 2.0 四个组成部分的提案需求(RFP),分别是:基础结构(Infrastructure)、上层结构(Superstructure)、对象约束语言(OCL)和图交换(Diagram interchange)的需求^[5]。其中基础结构和上层结构是 UML 2.0 提案需求的主体部分。四

个提案需求的关系参见图 1。

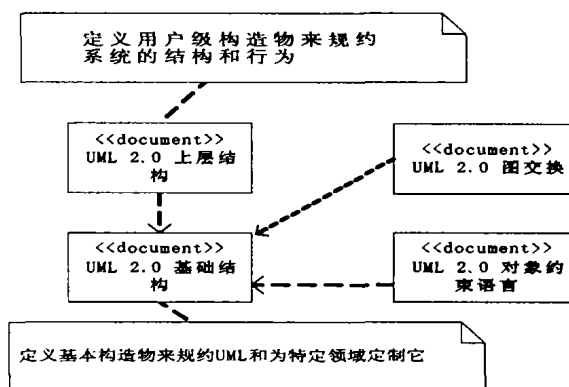


图 1 UML 2.0 的提案需求

这些提案需求确定了 UML 2.0 的主要目标是通过语言的重构和精化,使其更容易应用、实现和定制。基础结构提案需求是:提高 UML、MOF 和 XMI 在体系结构上的一致性;为 UML 提供两种扩展机制:基于元模型的一阶扩展机制和基于 Profiles 的元模型定制能力;更细粒度、灵活支持相容点(Compliance Points)的包结构。上层结构提案需求是:支持结构化建模,包括对基于构件的开发和实时构架的规约能力;提高行为建模(特别是状态机和交互)的封装性和可伸缩性;在活动图上增强对控制流和数据流建模的机制;对泛化、依赖和关联的语义精化等。对象约束语言提案需求是:定义一个与

^{*})本文研究得到国家自然科学基金(No. 60073015)、国家 863 计划(No. 2001AA113070)资助。马浩海 博士生,研究方向为计算机软件与理论,面向对象技术,建模语言与语义。邵维忠 教授,博士生导师,研究方向为软件工程与软件工程环境,面向对象的方法与技术,操作系统等。麻志毅 副教授,研究方向为软件工程,面向对象的方法与技术,计算语言学。

UML 元模型一致的 OCL 元模型;增强 OCL 的表达能力。图交换提案需求是:定义一个与 MOF 兼容的元模型,使得在描述图形元素和它们的布局时,和元类的语义联系起来。

UML 2.0 共经历了四年之久的规范制订的“奥德赛”过程^[6]。截至 2003 年 6 月 12 日,OMG 陆续采纳了 U2 Partners (U2P)的基础结构规范和上层结构规范,加上已经采纳的对象约束语言和图交换规范,UML 最重要的一次修订升级过程成功完成了。下面的内容首先概要总结 UML2.0 基础结构的设计目标和体系结构,之后简要介绍上层结构的设计目标和与 UML1.X 相比较的重要的修订特色,最后给出 UML

2.0 规范的反响情况,以及对其存在的问题的一些评述。

2 基础结构概览

UML 2.0 基础结构^[8]的设计目标是定义一个元语言的核心—基础结构库(InfrastructureLibrary),通过对此核心的复用,除了可以定义一个自展的 UML 元模型之外,还可以定义其他的元模型,包括 Meta-Object Facility(MOF)和 Common Warehouse Metamodel(CWM),见图 2(a)。由于共用核心包,所以 UML 和 MOF、CWM 在体系结构上更加一致。

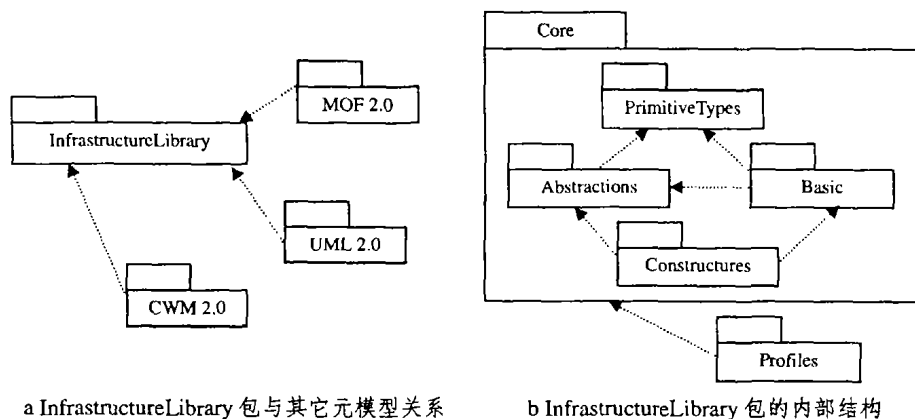


图 2 UML 2.0 基础结构

基础结构还提供了针对 UML 的两种扩展机制:(1)比 UML1.X 新增了一阶扩展机制(First-class extension mechanism)。通过重用 InfrastructureLibrary 包,可以增加新的元类和元关系,定义与 UML 相关的新的建模语言,从而形成语言家族。(2)通过外廓(Profiles)包提供定制 UML 的更强有力的机制,允许用户定义针对不同平台(如 J2EE,.NET)和领域(如电信,金融,系统工程)的方言。

图 2(b)给出了 InfrastructureLibrary 包的内部结构。InfrastructureLibrary 包由包含了高度可复用元模型构造物的 Core 包,和用于定义 UML 在不同领域的方言的 Profiles 包组成。Profiles 包依赖于 Core 包。出于复用的目的,Core 包进一步划分为一些子包:

原子类型(PrimitiveTypes)子包简单包含着一些在元建模时会用到的预定义类型。它们都是为 UML 和 MOF 特别设计的。其它元模型还可以添加其它的原子类型,或者重载这些原子类型。

抽象元素(Abstractions)子包主要包含着抽象元类,这些元类可以被进一步特化,或者被其它元模型复用。但是一般不建议元模型复用该包,出于这个原因,抽象包又被分成若干个更小的子包。

构造物(Constructs)子包主要包含着用于面向对象建模的具体元类。该包是为 MOF 和 UML 的复用而特别设计的,它是这两种元模型在体系结构上保持一致性的最重要的实现部分。

基础(Basic)子包包含着一些作为创建 XMI 的基础构造物,这些 XMI 可以用于任何基于 InfrastructureLibrary 的元模型,如 UML、MOF 等。

依赖于 Core 包的 Profiles 包,定义了裁减已有元模型,使其适用于特定的平台或领域的机制。裁减的对象主要是 UML 元模型,但是不仅限于此,任何基于公共 Core 包的元模型都可以利用 Profiles 提供的轻量级方法进行定制。

实际上,对基础结构而言,标准制订者和建模工具商的兴

趣要远远大于普通建模者。

3 上层结构概览

UML 2.0 上层结构^[9]的设计目标是:

- 严格地复用基础结构中的模型构造物;
- 既可以描述平台无关构件(如业务构件),也可以描述平台相关构件(如 EJB,COM+等),提高对基于构件开发和 MDA 的支持;
- 支持接口、部件和连接器,及其层次化组合能力,从而精化对构架规约的能力;
- 增强行为图的可伸缩性、精确性和集成能力;
- 支持可执行的模型;
- 审定所有 UML 1.x 的构造物和图,适当地精化、取缔和逐步淘汰。

上层结构中的用户模型构造物被分成三部分加以组织:第一部分“结构”,第二部分“行为”和第三部分“增补”。结构部分定义了静态的、结构化的构造物,如类、构件和组合结构等;它们主要被用在各种结构图,如类图、构件图和部署图中。行为部分定义了动态的、行为化的构造物,如活动、动作、交互、状态机等;它们被使用在各种行为图,如活动图、顺序图和状态机图中。增补部分定义了一些辅助构造物,如信息流、模型、模板、原子类型、以及在不同领域定制 UML 的 Profile 等。

上层结构规范在介绍模型构造物时,与 UML 1.X 一样给出它的抽象语法、约束和自然语言语义;和 1.X 不同的是同时附带了其符号表示法,而不是用独立的表示法部分再次描述每个模型构造物的图形符号;另外,增加了 1.X 中没有的语用信息,如语义变化点,向后兼容性和设计原理。出于用户理解的需要,2.0 规范中的使用举例也比 1.X 有了显著的增加。

上层结构是利用 UML 建模的用户最需要和最感兴趣的部分。下面我们对上层结构中 UML 1.X 相比较的主要修

订和扩展的部分进行介绍^[9,10,12~14]。

3.1 丰富的图

UML 2.0 支持 13 种图,与 UML 1.X 的 9 种图相比较,描述系统的视图和观点更加丰富。每种图的语用根据关注不同有所侧重,逻辑上可以归入两大类之一:结构图和行为图。但是 UML 2.0 将结构元素和行为元素结合在一起建模的方面有了很大的进步,如显示内部结构嵌套的状态机时,就可以将不同类型的图(如组合结构图 and 状态机图)混合在一起。因此,这两种类型图的区分并不是强制性的。

结构图包括:类图,组合结构图,构件图,部署图,对象图和包图;行为图包括:活动图,交互图,用况图和状态机图,其中交互图是顺序图、通讯图、交互概览图和时序图的统称。

与 UML 1.X 相比,组合结构图、包图、交互概览图和时序图都是新增的图。而原来的协作图改名为通讯图后,语义却没有被丰富到与顺序图等价的地步。状态图改名为状态机图,解决了 1.X 中状态图和状态机的语义重叠、模糊不清的问题。原来的集合词汇实现图被取消。

3.2 结构化建模

UML 2.0 引进了内部结构建模的概念。系统被表示为结构化类目,其操作表示了系统的功能,其属性表示了系统的各种特性。结构化类目通过组合结构图描述了共同执行一项系统任务的内部参与者的结构和关系。通过封装和隔离内部结构,实现对系统的可插拔替换能力。结构化类目的核心构造物包括部件(Part)、连接子(Connector),和端口(Port)。通过端口,类目可以对外发布两种接口:提供接口和请求接口,来封装类目和环境之间的通信。图 3 是一个结构化类目—自动售货机的例子。

结构化类目包括类、协作和构件,其中最具有代表性的是构件。一个构件是系统的一个模块化部分,它通过接口封装了内容并且在其环境中可替换。构件封装了内部所包含类目的行为和状态,对外通过提供接口和请求接口来定义它的行为。构件可以通过组装(Assembly),以及把它们的提供接口和请求接口线接(Wiring)在一起提供系统功能(见图 4)。在 UML 1.X 中,构件的概念主要用在设计阶段中的系统实现部分,

UML 2.0 将构件用在系统建模的各个阶段上面,最后在部署和运行时(Run-time)的环境中精化。

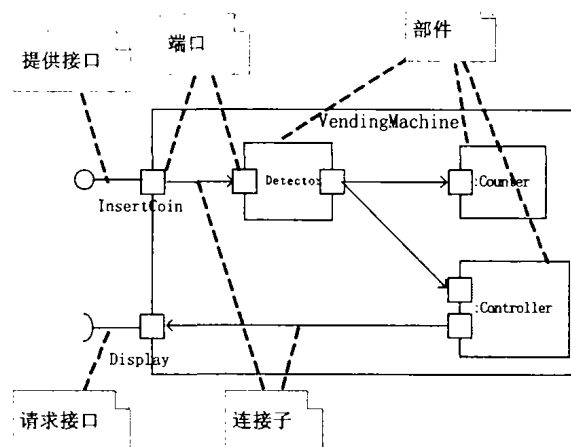


图 3 结构化类目-自动售货机

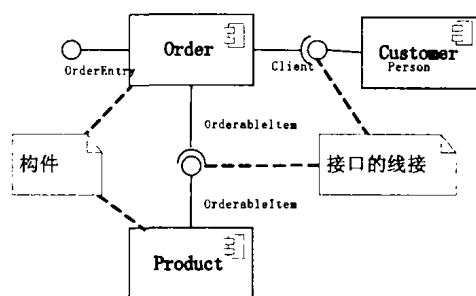


图 4 构件和接口线接

3.3 交互

交互侧重于实例间的通信,通过消息传递来表示操作调用和信号发送。依赖于使用交互的目的不同,一个交互可以用几种图形来表达:顺序图、通讯图、交互概览图和时序图。虽然每种图形提供适应不同情况的能力,但是 UML 2.0 中顺序图是交互图中语义最丰富、表现力最强的一种图。

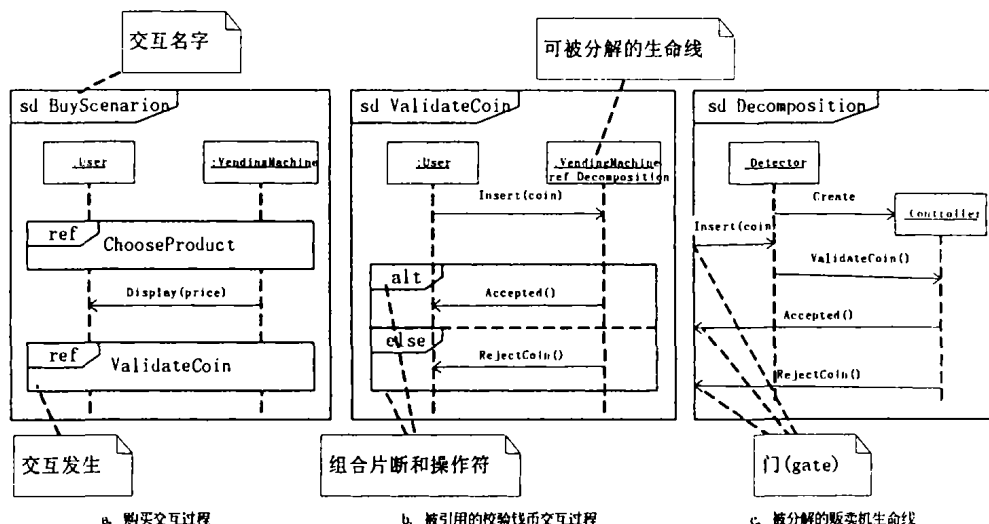


图 5 顺序图及新增交互描述能力

UML 2.0 中的顺序图在描述复杂交互的伸缩能力方面有了显著提高,新增了:①交互发生(Interaction occurrence),允许从一个交互引用到另一个交互,避免了交互的复制;②组合片断(Combined fragment)和交互操作符等,使得在顺序图中可以表示诸如选择、循环、并行、有序、引用等复杂的控制结

构;③对生命线(Lifeline)的分解能力(Decomposition),可以通过实例的内部结构来细化交互过程。图 5 是 UML 2.0 顺序图及新增构造物的例子。

新增的交互图之一是交互概览图,它是一个关注控制流,但是抽象掉消息和生命线的视图。它使用活动图的表示法,其

中的节点或者是一个交互或者是一个交互发生,见图6。纯粹的交互概览图中的所有活动都是交互发生,那么在图上就根本不会有任意的消息和生命线了。

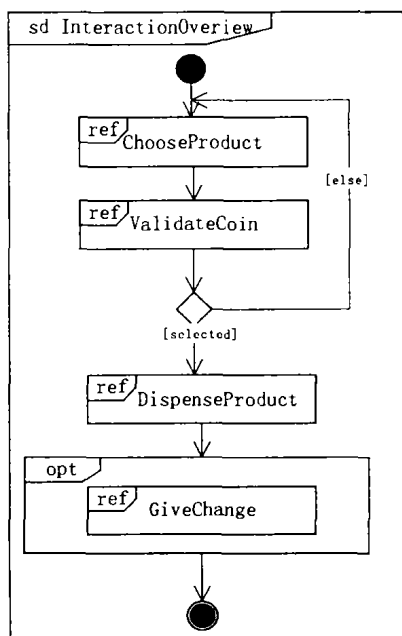


图6 售货机交互概览图

另一种新增的,特别适合实时和嵌入式系统建模的交互图称为时序图。时序图关注沿着线性时间轴,生命线内部和生命线之间的条件的改变。它描述对象状态随着时间改变的情况,很像示波器,适合分析周期和非周期性任务

3.4 活动和动作

活动建模的注意力放在低层协作行为之间的序列和条件上,而不是关注由哪个类目来执行这些行为。它们通常称为控制流和对象流模型。UML 2.0的活动图将活动和动作集成起来,增强了复杂过程流的建模能力与模型的可执行能力。UML 1.X中动作的数据/控制流模型由更通用的活动的流模型取代,而活动的行为调用由更通用的动作模型取代,这样活动定义了流图(过程),动作定义了执行行为的节点,使得行为建模更加直观有效。图7给出了一个模拟了更新客户帐户余额的活动图例子。

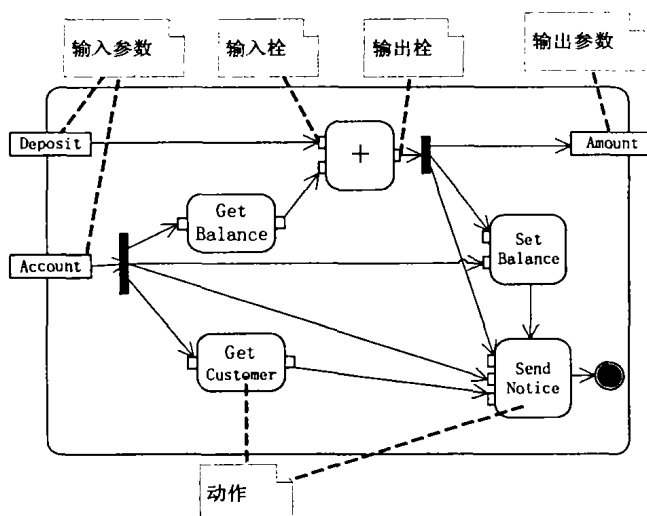


图7 更新帐户余额的活动/动作图

新增核心构造物包括:用作动作输入和输出的栓;结构化

节点和可中断区域等组合构造物。对原来的UML 1.X的核心构造物的语义充实包括:在边上增加了令牌,增加了流化和异常参数;增进了对活动图进行多维的、分层的、扩展的分割法;控制节点支持分叉、汇合、决策和合并等。

3.5 状态机

状态机通过有限状态变迁系统来规约离散的行为。UML 2.0对状态机的主要改进是:

- 通过状态机边界的出口/入口点,实现子状态机(状态机片断)的完全封装,实现可插拔替换能力。
- 状态机是可特化的。一个特化的状态机是一个泛化状态机的扩展,区域里的节点和变迁可以增加,节点和状态可以重定义(如:简单状态变成复杂状态,复杂状态增加状态和变迁等),变迁也可以重定义。状态机的特化可以依赖于它所关联类的特化而特化,也可以独立的特化。
- 通过协议状态机有效规约端口或者接口的约束操作序列。它通过变迁上的前置/后置条件描述了一个类目可以触发的合法变迁。

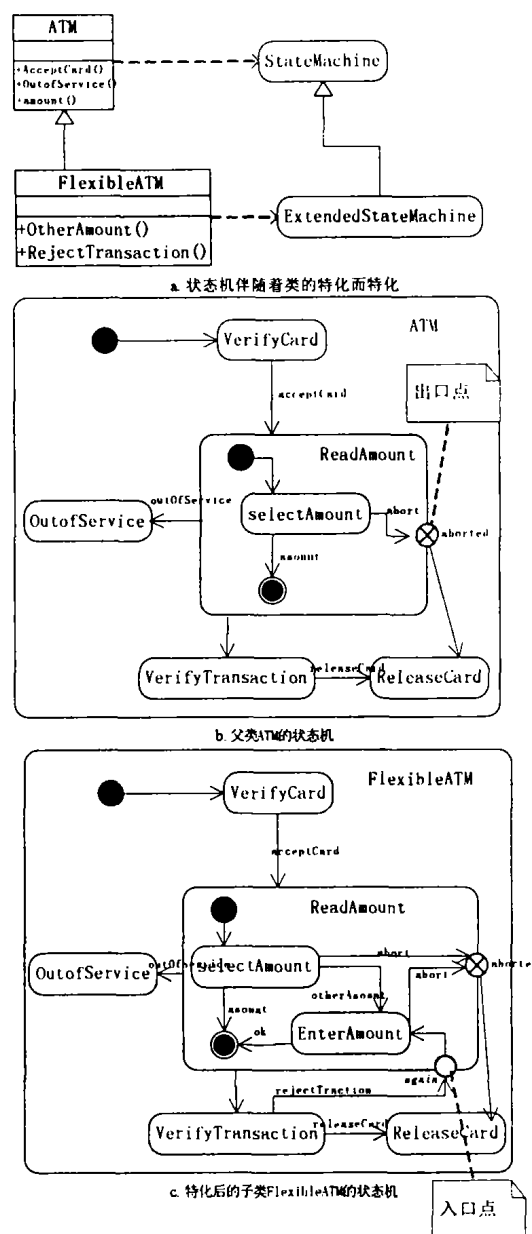


图8 状态机的封装和特化

图8给出了与类关联的状态机的例子。当类被特化时,子类关联的状态机也可以从父类关联状态机特化而来。

(下转第8页)

要取决于对系统的各项属性进行权衡的结果。一般地,这些设计决策都假设由设计者,而不是某些基于 QoS 的服务来决定。如果我们能够寻找出一种良好的权衡和评估方法,使得中间件框架能为应用提供这些决策帮助,将是一件非常有意义的事情。

此外,软件系统(或构件)在其运行过程中,如何在环境的变化下维护其可信属性?一般来说,支撑平台必须能够有机制和手段保证软件系统的可信度。例如,如果发生了网络攻击或者人为事故,使得软件系统某时刻某个服务不可用,支撑平台必须能够迅速作出反应(例如启动该服务的一个副本)以继续维持软件系统的可信属性。另外,如果软件系统发生了适应变化或者演化,支撑平台也应该能够维护它们的可信性。

结束语 本文以可信计算为研究焦点,描述了它在学术界和工业界的发展历程,并以此为基础探讨了在 Internet 环境下研究软件可信性保障技术所需要解决的问题,为下一步工作的展开奠定基础。

参考文献

- 1 Mundie C. Remarks on Trusted Computing Forum 2001, www.microsoft.com/presspass/exec/craig, Nov. 6, 2001
- 2 NSF/CISE-Trusted Computing, www.nsf.gov, March 2003
- 3 NSF Program Announcement/Solicitation: Trusted Computing,

- www.nsf.gov/pubs/2001/nsf01160/nsf01160.html#TOC, Dec. 2001
- 4 Department of Computer Science of the Technical University Darmstadt. Trusted Systems, www.dvsl.informatik.tudarmstadt.de/DVSl/research/index.html
- 5 Gates B. Trustworthy Computing, www.microsoft.com, Jan. 2002
- 6 Microsoft White Paper, Trustworthy Computing, www.microsoft.com, Oct. 2002
- 7 Microsoft White Paper. Building a Secure Platform for Trustworthy Computing, www.microsoft.com, Dec. 2002
- 8 Aderson R. TCPA/Palladium Frequently Asked Questions, www.cl.cam.ac.uk/~rja14/tcpa-faq.html, July 2002
- 9 Stallman R. Can you trusted your computer, newsforge.com search.pl? topic=19, Oct. 2002
- 10 Workshop on Trusted Computing Paradigm'02
- 11 Gates M. Motivation for TCPA and Descriptions of Some Components, mgates.oxonet.com/tcpa/report.html, May 2002
- 12 Pearson S. Trusted Computing Platforms, the Next Security Solution, www.informit.com, Aug. 2002
- 13 Pradham D K. Fault-Tolerant Computer System Design. Prentice Hall PTR, June 1996
- 14 Wu Jie. Distributed System Design. CRC Press LLC, 1999
- 15 Introduction to security, ou800doc.caldera.com/SEC-admin/CTOC-IntroScur.html, Caldera International, Inc., 2001
- 16 高信度计算的未来, www.ciweekly.com/article/20021111/2002111122259-1.xml, 2002. 11
- 17 Schneier B. Secrets & Lies: Digital Security in a Networked World, John Wiley & Sons, Aug. 2000

(上接第4页)

4 评述

UML 2.0 被采纳之后,大多数人认为这是一个更加平衡的语言,特别是在行为图的集成能力,对构件、实时和嵌入式系统、业务过程的描述能力方面有很大的增强^[15]。但是在文[6,7,11,16]中提到的对 UML2.0 的担忧有些仍然存在,表现在:

1)没有给出 UML 一个简洁、精炼的核心。相反,通过 InfrastructureLibrary 定义了一个语言家族的核心。通过对其应用复用、特化、重定义等办法来定义诸如 UML 和 MOF 等其它元模型。而多数用户使用的 UML 本身仅仅体现在上层结构中。上层结构的规模仍然庞大繁杂,没有做到使用 UML 语言的 20%来详述 80%的常见软件问题^[7]。

2)四层元模型的体系结构被弱化。UML 和 MOF 更多地站在了同一个层次上,都是 InfrastructureLibrary 的实例。对于已经熟悉了 OMG 四层元模型体系结构的用户,需要重新来理解新的、基于语言家族的体系结构了。

3)UML 2.0 由相对独立的四个规范组成,并且大量公司参与了制订过程,使其不可避免地出现了“第二系统病症”问题^[17]。与 UML1.X 相比,UML2.0 的建模元素从 170 多个增加到了 360 多个,图从 9 种增加到了 13 种,规范文档从 700 多页增加到了总计 1000 多页。虽然,UML 2.0 试图通过包内相容点(compliance point)的定义使得用户和工具商可以递进地、有选择地实现语言特性,但是,毕竟掌握和学习 UML2.0 变成了一件很不轻松的事情。

4)工具商在初期会因 UML 2.0 过于庞大复杂的规模,不得不按照规范中的包内相容点来有选择地实现 UML。但是之后呢,出于市场竞争的需要,工具商可能又会发展到对整体规范的完全支持,就像 SQL 语言发展的老路一样。这样的结果,对用户从可用性、经济的角度来说造成的影响还有待于时间的检验。

5)新增的一阶扩展机制允许用户定义新的模型元素,甚至是语言。自由地定义各种不同于 UML 的新的建模语言究竟是拯救了 UML 还是重新开始一场语言之争,其真实效力如何还有待于实践的检验。

参考文献

- 1 Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language User Guide. Addison Wesley Longman, Inc. 1999
- 2 Cook S. The Architecture of UML. OMG presentations, 2002
- 3 Salicki S, Jourdan J, de Miguel M. Practical Experiences in the Application of MDA. UML 2002
- 4 Weigert T. UML 2.0 RFI Response Overview. Motorola, Dec. 1999
- 5 Kobryn C, Weigert T. OMG UML 2.0 RFPs ad/00-09-05. Object Management Group, Sep. 2000
- 6 Kobryn C. UML 2001: A Standardization Odyssey. Communications of the ACM, Oct. 1999,42(10)
- 7 Kobryn C. Will UML 2.0 Be Agile or Awkward? Communications of the ACM, Jan. 2002,45(1)
- 8 UML 2.0 Infrastructure, 3rd Revision. OMG document ad/03-03-01. Object Management Group, 2003
- 9 UML 2.0 Superstructure, 3rd Revision. OMG document ad/03-04-01. Object Management Group, 2003
- 10 Unified Modeling Language Specification, Version 1.5. OMG document formal/03-03-01, Object Management Group, 2003
- 11 Kobryn C. Designing a More Agile UML 2.0. Telelogic Research Center, 2002
- 12 Björkander M, Kobryn C. Architecting Systems with UML 2.0. IEEE SOFTWARE, July/August 2003
- 13 Kobryn C, Björkander M. UML 2.0 Roadmap: What Should Systems Engineers Expect? UML for Systems Engineering Workshop, Chicago, USA. Sep. 2002
- 14 Douglass B P. UML 2.0: Incremental Improvements for Scalability and Architecture. I-Logix Co. 2003
- 15 Vaughan J. Are you ready for some UML 2.0? Application Development Trends, Sep. 2002
- 16 邵维忠,杨芙清.面向对象系统设计.清华大学出版社,2003
- 17 Brooks F. The Mythical Man-Month, Anniversary Edition. Addison-Wesley, 1995