

# 可执行 UML 建模技术研究

黄毅杰

(漳州职业技术学院, 计算机工程系, 福建, 漳州, 363000)

**摘要:**能够降低软件开发成本、提高软件生产效率、可移植性、增加软件的可重用性的技术越来越受到人们的关注,通过可执行 UML 技术,使得软件开发在前期可以得到执行和验证,通过可执行 UML 的动作语言可以解释对象之间的行为等。

**关键词:**可执行 UML; UML; 平台无关

**中图分类号:** TP37 **文献标识码:** A

## 1 引言

UML 是模型驱动架构的一种关键技术,每一个由模型驱动架构构建的模型都是基于一个平台无关的 UML 模型。但是 UML 的语义不是十分精确, UML 在软件分析设计阶段是不可以执行、不能验证的,增加了后期开发的风险, UML 无法直接表示数据处理过程,如对象与对象是如何产生实际的连接等。为了解决建模的这些问题, OMG 对 UML 进行了扩充,在 UML 中加入了动作语义使得模型可以是可执行的。

## 2 可执行 UML 概述

可执行 UML 规定了一个简单、一致的 UML 符号的子集。这些符号的选择是基于现实世界中的结构,而不是基于在某个软件系统的构造中用到的个别结构。不仅仅这些符号本身是简单的,而且组织和集成它们的方式也必须遵循严格的规则,而这些规则能够保持整个系统规约的清晰性。

核心 UML 加上动作语义就成为可执行 UML[1](Executable UML)。可执行 UML 用动作规约语言对 UML 进行扩充,使得模型可以被更精确地描述,得到可执行的平台无关模型 PIM,同时可执行 UML 去掉了庞大的 UML 体系中语义较弱的部分,解决了 UML 歧义问题,使得建模过程更加简洁明了。可执行 UML 就是 UML 的一个可执行版本<sup>[2]</sup>,它还包括了:

一个被清洗定义的简单模型结构;

精确地动作语义,这些语义已经成为 UML 标准的一部分;

一个适应性很强的动作规约语言;

一个配套的关键过程 MDA,该过程面向可执行建模,大规模复用,基于模式的设计。

## 3 可执行 UML 建模过程

### (1) 系统用例建模

用例仅从用户使用系统的角度描述系统中的信息,并不描述系统内部对该功能的具体操作方式,确定系统能够做什么?谁来使用这个系统?用例它描述了待开发系统的功能需求,它将系统看做黑盒,从参与者的角度来理解系统,它驱动了需求分析之后各阶段的开发工作,不仅在开发过程中保证了系统所有功能的实现,而且用于验证和检测所开发的系统。建立用例的过程是一个不断细化的过程,刚开始建立的用例可能比较粗糙,随着对需求的不断挖掘和理解,逐步细化用例。

### (2) 域的划分

系统分解的基本单元是域。域表示一个大的可复用组件,采用由 UML 包图和依赖关系构成的域图来描述。建立一个系统涉及许

多不同的主题并把它们连贯起来形成一个整体。每个主题都是一个域,是能够被理解和使用可执行 UML 建模的。我们可以为每个域建立一个或多个可执行 UML 模型。

域的类型有可以分为以下四种它们分别是应用域、服务域、体系结构域、实现域。

### (3) 使用域进行平台无关建模

建立一个域的平台无关模型将在 3 个层次中构建:

第一层次是对域中的类建模。

平台无关模型的主要组成部分是类图,每个域都有一个类图。类图是系统的静态观点。它描述了类和关系。类图描述了被分析的系统的抽象表示,以及这些抽象表示之间的关联关系。类图不描述什么时候创建或删除一个类或关系的实例,也不描述如何使用对象或如何查询关系。换句话说,类图是声明性规约。对于域类图的正确答案不是唯一的,对于任何域,都有多个可能的类图能够满足相关要求。

第二层次是定义域的动态行为。

模型需要定义类之间如何互相实现域中要求的行为。在可执行 UML 中提供了两种状态机的表现形式,一种是状态图,状态图显示了状态、事件、和转换,提供了状态机的图形表示,但图不能涵盖所有可能的组合;一种是状态转换表(state transition table,简称 SST),使用状态转换表确保底层模型的完整性。一个是代表性的图形,但不完整,另一种是表格,显示状态和事件的所有组合。在状态转移表,每行代表一个状态,每列表示一个事件。这些单元格指定什么时候会发生在给定的状态(行)检测到一个特定的事件(列)的对象,这些单元格被称为效果。

第三层次是详细定义动作,即用动作语言来表示。

动作语言是平台无关的语言,用来在可执行 UML 模型的上下文中详述处理行为。该语言的目的就是为将要被系统执行的处理行为提供无歧义的、精确的且易读的语义。UML 并没有定义动作语言,而是在 UML 标准中加入了动作语义。而动作语言符合 UML 的动作语义。

### (4) 验证 PIM

我们可以用模拟器来验证每个域的 PIM 是否表现出我们所要求的行为,最后,我们通过集成一系列相互兼容的域来完成系统的集成。

### 4 可执行 UML 的动作语言

可执行 UML 中关键的一环就是动作语言<sup>[3]</sup>,动作语言能够精确描述对象行为,能够使建立的模型是可执行,动作语言是抽象的、独立于各种程序编程语言(如 C++, Java 等)。在

国外,目前已经有几个工具厂商推出了可执行 UML 的动作语言和用于系统开发的 CASE 工具,通过这些工具可以对模型进行执行和验证。下面介绍几种运用比较广泛的动作语言:

### (1) Object Action Language(OAL)

这个语言由 Project Technology Inc. 公司发布, OAL 被用于定义模型中过程的语义。在由 Project Technology Inc. 公司发布的 BridgePoint CASE 中, OAL 可被用于以下几个模型元素中:对象状态、定义两个域模块之间的桥程序、对象的函数、基于类和对象的操作和基于对象属性的操作。

### (2) Action Specification Language(ASL)

动作规约语言 ASL 是一个与实现语言无关的动作语言,通过它可以使模型成为可执行和可验证。这个语言由 Kennedy Carter 公司发布。ASL 是满足 "Precise Action Semantics" 的,它是 UML 标准的一个扩展。当 ASL 被执行时,模型变得更加容易理解。

### (3) SMALL

SMALL 是 Shlaer-Mellor Action Language 的缩写,最后一个 L 是没有意义的,这个语言是由 Stephen J. Mellor 创造的。Gregory Rochford 和 Cortland D. Starrett 建立了一个解析器来验证 SMALL 语法。

### (4) TALL

TALL 是 That Action Language 的缩写,最后一个 L 没有意义。这个语言是由 Marc J. Balcer, Conrad Bock 和 Dirk Epperson 共同创造的。

结束语:可执行 UML 解决了 UML 在应用过程中暴露出的各种弊端如 UML 在软件分析设计阶段是不可以执行、不能验证的,增加了后期开发的风险,信息表达度问题, UML 无法直接表示数据处理过程,如对象与对象是如何产生实际的连接等。近几年来,基于可执行 UML 的软件开发方法已成为软件开发方法的一个研究方向。

### 参考文献

- [1] 刘建宾, 李建忠, 余楚迎. 模型驱动体系架构 MDA 及 xUML 规范在其语境中的探讨[J]. 汕头大学学报(自然科学版), 2004 年 11 月 58-65.
- [2] Chris Raistrick, Paul Francis, John Wright, Colin Carter, Lan Wilkie 著. 赵建华, 张天等译. MDA 与可执行 UML[M]. 机械工业出版社, 2006.4.
- [3] 兰庆国. 可执行元模型关键技术研究[D]. 吉林大学博士学位论文, 2006 年 4 月 66.