

对象建模技术

马希荣 (宁夏大学计算中心)

一、对象建模技术的概述

对象建模技术(简称 OMT)是一种面向对象的软件开发方法学,它与传统的面向功能的方法学截然不同。在传统的方法学中,强调的是区分和分解系统的功能,这种做法虽然是目标的最直接的实现方式,但生成的系统可能是无效的,假如系统需要改变,功能分解系统则需要做大的变动。

模型是为了解事物而对事物作出的一种抽象,是在实现问题求解之前,为理解问题而建造的一种对象描述。模型忽略了事物不必要的细节,因而对原始实体的处理会更加容易。许多年前,工程师、艺术家就是采用构造模型来试验自己的设计如为风道试验而建立的机翼模型和书稿草图等。计算机软件系统的开发也不例外。模型是真正建立系统前进行物理实体的测试,它能提前发现错误并予以修正。它可以与用户交流,将构造的模型显示给用户看。模型是模拟了系统的部分或全部的外部行为,模型的主要目的是减少系统的复杂性。

对象建模技术包括三种模型,即对象模型、动态模型和功能模型。对象模型是通过描述系统中的对象、对象间的关系,标识类中的对象的属性和操作表组织系统的静态结构,是三种模型中最重要的模型,它为动态模型和功能模型提供了实质性框架。动态模型描述与时间操作和次序有关的系统属性,它关心的是控制,从事件和状态的角度描述系统的数据结构。功能模型从值和功能的角角度描述系统的计算结构,它不考虑系统干什么或如何干。一个典型的软件肯定包含了这三种性质:它使用数据结构(对象模型),顺序执行操作(动态模型),并且完成值的变化(功能模型)。这三种模型描述的是系统的不同方面,但各种模型之间又互相联系,对象模型描述了动态模型和功能模型中操作的数据结构,对象模型中的操作对应于动态模型中的事件及功能模型中的功能。而功能模型描述了对象模型中的对象唤醒和动态模型中行为唤

醒的功能。

二、面向对象的建模技术

1. 系统分析

一个好的分析模型应能反映问题所有的本质属性,而不引入实现的考虑。首先,着手于对问题的初始陈述,同用户、需求者及应领域的专家磋商这种陈述,需求应该描述需要什么,而不是如何。问题陈述可能是不完全的、模糊的或错误的,它仅仅是起点。

一个现实问题可以更自然更实际地被看成一组对象及其相关方法的集合。面向对象设计的初始元素是对象。以后,随着对通用性的标识,对象聚集成类,然后又进一步子类化成更抽象的类,最高层的抽象称为一个框架。一个框架就是一个类集,这些类表达了对一组相关应用程序的设计。从实质上讲面向对象的问题求解是从已有的类库开始并重用它们,根据系统要求剪裁这些通用类。这种方法就需要提供标识定义和组织类及其方法和消息的规则,还需要能提供组织类库的策略和从已有的框架或类库建造应用程序的准则。

(1)构造对象模型。分析需求的第一步是构造对象模型,对象模型表示了客观系统的静态数据结构,它描述了现实世界的对象、类及它们之间的关系。其信息来源于问题陈述,应用领域的专业知识和客观世界的一般性知识,初始的分析模型很中能包含错误,这些错误必须在以后的反复分析中加以更正。在构造对象模型时,首先确定对象、类;然后确定对象之间的关联,包括聚集。也要确定对象属性和链。使用继承来组织和细化对象、类。将紧耦合的类和关联的构造成模块,可以用简洁的文字,对对象模型中的信息加以补充。

(2)开发动态模型。动态模型表示了系统的行为及其对象。在分析阶段不考虑算法的执行,算法是实现模型中的一部分,对于静态数据存储器来说(如数据库),动态模型毫无意义。动态模型对相互作用的系统具有很重要

的意义。

构造动态模型时先准备典型情况的脚本,然后确定系统和外部事物之间的外部文件,为各个主对象建立状态图,以表示对象接收和发送的文件模式及其文件执行的动作,状态图的结果集合构成了动态模型。

(3)构造功能模型。功能模型是用来说明值是如何计算的,而不考虑动作顺序。它表明了值之间的依赖关系及其相关的功能,功能模型的构造按如下步骤进行:

- ①确定输入、输出值;
- ②建立数据流图(DFD);
- ③描述函数
- ④确定约束条件;
- ⑤确定优化标准。

分析的最后阶段,还要验证、重复并完善,细化上述三种模型。

2.系统设计

系统分析完毕就进入系统设计。系统设计是问题求解及建立解答的高级策略。它包括将系统分解成子系统的策略,子系统的硬、软件配置,详细设计框架的构造等方法策略。

系统设计要制定解决问题的基本方法。将系统分解成子系统的全面组织结构。目的是使多个设计者能各自独立地进一步设计各个子系统,应当制定以下策略:

- (1)系统分解。可将系统分成水平层次和垂直块。系统自身也就是层次和块的混合物。
- (2)确定固有的并发性;
- (3)处理器任务的分配;
- (4)数据存储的管理。可以用内存数据结构、文件、数据库来实现。
- (5)全局资源的处理
- (6)选择软件控制机制
- (7)边界条件的处理
- (8)设置折衷优先级

3.对象设计

对象设计时对分析模型进行详细分析和阐述,并且奠定实现的基础。它要制定出实现系统所必须的策略但不涉及语言的具体细节。分析阶段确定所实现的内容,系统设计阶段确定实现策略,而对象设计则确定对实现要用到的类,关联的完整定义,实现操作的算法。对象设

计与传统结构的初步阶段相似。实现对象设计的步骤如下:

(1)将三种模型结合起来。在分析阶段我们得到了对象、动态和功能模型。其中对象模型是组织对象设计的主要框架,将动态模型中的动作及活动以及功能模型中的处理转换成操作,加入到对象的类中。

(2)实现算法的设计。功能模型中的各个操作都要用算法来表示。算法是用来说明如何执行操作的。算法设计首先要选择最小开销的算法,要考虑到计算复杂度,易理解,易实现这些因素。在选择算法时,涉及到选择算法使用的数据结构的问题。在对象设计阶段,必须为有效的算法选择其数据结构形式。常见的数据结构都是包容类的实例。如数组、表、队列、栈、集合、字典、关联等,定义新的内部类和操作,在展开算法时,可能需要引入一些新的对象、类,用来存放中间结果。在分解高层操作时,也可能引入一些新的底层操作。

(3)优化设计。分析模型概括了系统的逻辑信息,但设计模型必须加入详细内容以支持有效的信息访问。效率低但语法正确的分析模型应当进行优化,目的是实现更为有效。优化时要考虑增加冗余关联,以减少访问开销,重新调整执行次序以提高效率。定义新的对象或新类来隐藏或存储在计算形式中,避免重复计算的开销。

(4)实现系统设计中的软件控制

(5)调整继承。随着对象设计过程的深入,经常调整类及操作的定义以提高继承的数目。

(6)设计关联的实现。关联是对象模型的纽带,它提供了对象之间的访问路径。

(7)确定对象表示。对象大多数是直接实现的,但设计者必须决定何时使用基本类型来表示对象,何时将相关对象组合起来。

(8)物理组装。程序必须组装成物理模块,将类、关联封装起来。

4.实现

在实现中,为了直接跟踪系统,为了使具有最好的灵活性和可扩充性,必须制定和遵循一套良好的设计准则。实现 OMT 设计最自然的方式是采用面向对象的语言,如 C++。因为面向对象语言的构造与 OMT 设计的构造是很相似的。当然也可以使用非面向对象语言或数据库来实现 OMT 设计。