

文章编号:1001-5051(2005)02-0164-04

面向对象分析建模策略及相关讨论^{*}

贾¹, 钱乐秋²

(1. 浙江师范大学 计算机科学研究所, 浙江 金华 321004; 2. 复旦大学 软件工程实验室, 上海 200433)

摘 要:首先介绍了面向对象的软件分析建模的重要性;并在分析比较目前流行的几种面向对象分析建模策略的基础上,着重介绍了在开发青鸟系统系列工具中采用的一种基于面向对象技术的分析建模策略;对分析建模过程中涉及的一些关键问题进行了探讨.

关键词:面向对象;面向对象分析建模策略;主题;青鸟系统

中图分类号:TP311 **文献标识码:**A

0 引 言

开发一个大型的软件系统并不是一件容易的事情,特别是青鸟(Jade Bird)系统的开发.开发成败的关键是必须清楚地了解用户的需求,只有这样,才能设计出符合用户的要求、容易使用、修改和维护的系统.因此,有必要建立一种帮助认识需求的模型,使用户和开发人员之间进行充分的交流并达成一致,在此基础上逐步细化设计软件系统的结构,进行概要设计和详细设计并最终实现软件系统.

在整个软件开发中,完成从问题空间到方法空间的转换,并奠定了后续工作的基础,被公认为软件开发中最困难的、也是亟待解决的一个问题.分析模型是在领域需求分析的基础上构造的,它的建立依赖于特定的建模方法,在信息系统中可以用实体-联系的方法(即 E-R 图法)来描述,其方法在一般的软件工程书籍中均有介绍.但是,用此方法只能反映概念间的静态关系,如果要体现其动态关系,更好的办法是采用面向对象技术.面向对象技术是一种能为软件开发提供更容易理解的需求说明、更清楚的设计和更容易维护系统的建模技术.文献[1~3]在软件开发时,着重考虑的是系统的开发 and 设计,忽略了如何去分析构造一个模型,以及在建模过程中可能会涉

及到的问题.笔者拟在这些方面进行探讨.

1 面向对象分析建模策略

典型的面向对象分析策略有:Booch 方法、Shalear-Mellor 方法、Coad & Yourdon 方法、OMT 方法等.

1.1 Booch 方法

Booch 方法^[1]是一种基于词法分析的方法.建议分析人员从目标系统的一个平凡描述开始,形成一个正文形式化的策略,指出正文中的名词、动词和形容词,然后由名词导出对象,由形容词导出对象的属性,由动词导出对象的操作,最后建立对象的界面(操作).该方法认为,针对问题的分析设计应是反复的渐增的过程,包括 4 个步骤:确定类与对象;确定类与对象的语义;确定类与对象的关系;确定类与对象的属性.该方法从动态、静态、逻辑、物理等多个角度描述系统,主要文档有:类结构图、对象结构图、模块结构图、状态转换图、时序图等.与其他的 OOA 方法相比,显得简单而实用;同时由于其在 OOA 中的奠基性地位,因而至今仍在众多的 OOA 方法中占有一席之地.Booch 方法对于图的使用没有明确的指南,只凭分析设计者个人的发挥,在实际应用中须确立详细的步骤,因而很难适应复杂系统的分析.

1.2 Shalear-Mellor 方法

^{*} 收文日期:2003-03-14;修订日期:2004-12-28

作者简介:贾 (1965-)男,浙江金华人,副教授,硕士.研究方向:软件工程;智能计算.

Shalear-Mellor 方法^[4]又称为 OOSA 方法(Object-oriented System Analysis),由 Shalear 和 Mellor 等人于 1988 年借助信息建模方法提出的一种面向对象的系统分析模型方法。该方法的步骤有:建立信息模型;确定对象生命周期;建立动态关系;建立动态系统;建立过程处理模型;确定范围(域)和子系统。开发中分别采用 ERD、DFD 和 STD 表示对象的结构、操作和生命周期,通过子类建立对象关系网络。由于该方法是数据库模型化技术语义模型的发展,因而与语义模型相似,继承了语义模型详尽的关系定义与属性定义。该方法隐含类、继承和封装这 3 个基本原则,只限于分析阶段,对设计阶段缺乏支持,适用于实时系统开发。

1.3 Coad-Yourdon 方法

Coad 和 Yourdon 于 1991 年提出了一种演进式 OOA 方法^[2],并用以下等式概括了其基本思想:

OO = 对象 + 分类 + 继承 + 通过消息通讯。

Coad 和 Yourdon 方法建立在对象、属性、服务等日常概念的基础上,致力于对问题的理解,从而建立有关系统的领域模型,并由此进行演化,力图完整真实地反映用户的需求,最终导出系统模型。其需求分析由 5 个层次构成:类及对象;结构;主题;属性和服务。相应地,分析过程也有 5 个步骤,每个步骤的进行都将进一步完善需求契约的信息。这 5 个步骤为:标识应用领域中的关键类及对象;刻画类及对象之间的聚合关系和继承关系得到结构标识;将类及对象组织成由若干主题构成的主题层;定义属性,借助某些问题来识别和详细说明系统中类及对象的属性;定义服务,即详细说明系统中类及对象的服务,用服务流程图的结构形式来刻画其处理逻辑。这种方法由于其理论体系比较完善,并经过一定的实践检验,对于一般的中小规模的系统比较合适。但其缺点是缺少对功能语义、全局控制的支持,应用 OOD 时需加以改正。

1.4 OMT 方法

Rumbaugh 的 OMT(Object-oriented Modeling and Technology)方法^[3],借鉴了 Shalear-Mellor 方法的模型思想,采用对象模型、动态模型和功能模型描述系统,支持软件工程的整个生命周期的开发。从 3 个不同的但又相关的角度出发进行系统的建模,3 个模型各自用不同的观点全面反映系统的需求。对象模型描述系统中对象静态结构及对象间的关系,动态模型描述系统中与时间变化有关的性质和对象交互行为,功能模型描述系统中数据的变化过程。它包括分析、系统设计、对象设计几个阶段。与其他方法相比具有 2 个优点:图表和模型规定详细,描述能力强;工作程序具体。从某种意义上说它是偏向分析的方法,但仅凭预备的文档,对于大型系统的描述

是不够充分的。

2 基于面向对象的分析建模策略

国家“八五”重点项目“软件工程开发环境(CASE)标准化与实用化”JB 系统的一些实用工具,如结构化分析和设计工具(SAT/SDT)、需求文档分析工具(RDAT)等已通过鉴定,并在 JB 中得到进一步的完善。在实践中,所采用的策略是一种基于面向对象技术的分析建模策略。

2.1 提出的理由

在文献[2]中,Coad 和 Yourdon 把主题描述为:主题是控制读者在同一时间内所能考虑的模型规模的机制,也是模型中一组意义紧密相关的对象类和结构的合成体,反映了系统的一定的侧面。主题与“子系统”概念类似,其目的是提供一种便于理解系统顶层视图的手段。主题除了是控制读者模型规模的工具,还应是模型的构造者控制模型复杂度的工具。面向对象技术有助于理解问题,但问题所反映的复杂度并不会因此而自动消失,需要一些辅助的手段来控制在同一时间内所能理解和解决的问题规模。由于目前的面向对象方法一般都只适用于中小型的系统开发,其中一个重要的原因是它们都没有提供完善的复杂度机制。抽象在面向对象技术中起着重要的作用,但通常的对象类的抽象不足以应付有一定规模的系统开发。

主题之间可以建立消息连接;不同主题的对象类之间,除消息连接外还可建立实例连接和组织结构,但不考虑分类结构。理由是使用实例连接有助于澄清消息连接的具体含义;使用组织结构有助于在系统开发中组织分工协作和以受限的方式引用他人的工作;不用分类结构是因为一个主题中的对象类的内部细节应尽可能局限于该主题内,而通常分类结构中父类的内部细节是向子类公开的。主题层相对稳定,对象层其次,结构层也较为稳定,最易发生变化的是服务层和属性层。因而在进行面向对象分析建模时,建议首先快速标识出主题,并按照可见性及引导读者注意的原则,组织主题的层次,这对于抓住系统的实质具有重要的意义。在有了初始的主题后,针对这一主题标识出与本问题有关的对象类,并随着对问题的深入理解再标识出其他的主题。主题在对象类之上提供了更高一级的抽象。

2.2 基于面向对象技术的分析建模策略

分析建模方法所使用的主要术语和记号类似 OOA^[2],但进行了改进。该方法同样基于 5 层模型:主题层、对象层、结构层、服务层和属性层,强调的是对领域的总体把握,使得系统开发的活动的角色以对象的角色为中心,能

更好地遵循 OOA 的封装性原则. 具体的实施步骤为: 首先, 由有经验的分析员根据用户的问题陈述、领域知识、以往的工作经验和常识, 快速地标识出主要的初始主题集, 以后的工作就从这一主题开始. 当然, 随着对问题的深入理解, 还会标识出其他的主题. 其次, 在有了初始主题后, 针对每一主题标识出相关的对象类, 标识对象类的方法可采用罗列名词词组产生候选的对象类进行筛选, 也可用 Lubars 等人的构造对象模型方法^[5].

接下来的工作就是要建立对象类空间的结构, 通过继承关系和聚合关系标识出分类结构和组织结构, 分别表示出问题空间的一般/特殊和整体/部分关系. 最后确定出每一对象类的作用, 即对外所要提供的服务及为完成这些服务而必备的辅助服务, 并在此基础上确定每一对象类必备的属性. 实施步骤为: 标识主题 标识对象类

标识结构 定义服务 定义属性. 以上的描述是线性的, 而在实际的应用中则必定是反复的, 需要经过反复的评审, 才能建立系统的用户模型, 同时各步骤的顺序只是指导性的.

3 相关讨论

3.1 标识对象

面向对象分析模型的核心是对象, 标识对象的目的是使一个系统的技术表示同现实世界概念联系得更加紧密, 产生一个稳定的框架模型, 便于考虑问题空间并收集用户需求, 避免从分析到设计时改变系统的基本表示. 标识对象可采用的策略有: Booch 方法、Coad 方法及 Lubars 方法. Booch 方法是一种基于词法分析的方法, 它从目标系统的一个平凡描述开始, 在此基础上标识潜在的对象. 虽然对于实际的复杂系统, 要考虑的问题比找名词要复杂得多, 但这种简单而实用的方法为分析模型中的对象标识奠定了基础. Coad 方法是通过在一定的范围内寻找候选的对象, 从中提取候选的对象, 并应用有关的规则对候选的对象进行筛选, 形成可用的对象. 该方法对于一般的中小规模的系统是比较合适的, 这是由于该方法的理论体系比较完善, 并已经过一定的实践检验. Lubars 方法的前提条件是需要有一个用户需求文档, 其策略是为用户需求文档中的每个句子建立一个对象或动态模型分片. 相比而言, 这种方法更适用于较大的问题, 因为它工作于段落和节的层次而不是词和词组; 此外, 其结果更接近需求模型, 因为基于图形的子模型包含了真实需求模型同样的构造, 即对象模型和状态转换图.

在实际的开发中, 对象的标识可用几种方法结合进行. 在青鸟系统的开发中, 通过对问题域的调查研究, 征求问题域专家的意见, 参阅其他类似的系统以及阅读有

关资料(包括七五期间的文档), 先用 Booch 方法标识出候选的对象, 再用 Coad 方法的有关规则进行筛选, 形成可用的对象. 在提取候选对象时, 也可结合 Lubars 的对象模型分片策略.

3.2 开发规范

一般的面向对象分析建模方法存在一些缺陷, 如 OMT 记号系统显得复杂和繁琐, 描述的对象模型也不如其他方法对象引用表、客户服务图清晰, 为此可在分析建模方法中采取一些措施, 如引进对象描述卡和改进记号系统等. 此外, 辅助图表由于简单有效, 可使开发过程清晰、易理解和实现. 在 JB 系统系列工具的开发中, 借助了大量的辅助图表, 同时为了使各开发单位之间增加协作与交流, 还专门制定了一整套的规范来给出系统的一致性表示, 以有效地管理和控制整个软件开发过程. 在 JB 系统最后提交的文档中, 包含了类及对象描述卡、对象引用表、客户服务图及对象界面图. 其中对象描述卡内容包括对象类名、类及对象的概要说明、关系说明、属性说明、提供的服务与描述、引用的服务等. 客户服务则通过考察每个类是否需要别的类服务及需要哪些类的哪些服务得到, 并从中得出对象引用表和对象界面图.

3.3 建模过程

开发初始的系统分析模型是一个困难的任务, 但是只要系统的总体描述或详细的客户规格说明书的存在, 采用自顶向下方法和自底向上方法一般都能完成建模的过程. 在这 2 种方法中, 自底向上方法可确保系统的正文文档中的所有需求都将得到重视; 自顶向下方法则可提供将各个需求紧密联系在一起的分片框架. 在实际中, 若能将 2 种方法结合在一起使用, 可能会比单独使用其中任何一种方法更为有效. 孤立地采用自底向上方法和自顶向下方法是困难的. 例如, 在自顶向下分析的脚本描述的具体行为序列就必须同系统行为的一般描述结合起来(其本身是一个自底向上的活动). 同时, 针对实际问题的分析建模过程是反复的、递增的过程. 一般地说, 自底向上方法的分片驱动方式适合于所提供文档材料的量大且丰富的情况, 因为它提供了完整性和可跟踪性. 如果没有目标系统的高层次信息, 把分片合成为一个单一的、连贯的模型是困难的, 而对于自顶向下方法的脚本驱动方式, 由于没有系统的规格说明, 反而是合适的. 在这种方法中, 如果没有有关目标系统的信息, 只是从领域专家导出系统的操作脚本, 则可通过提出问题来确保脚本的完整性.

另外, 还可能存在这样一个问题, 即模型中的对象数目会不会超出人的处理能力范围. 当系统较大时, 有必要结合自上而下的方式, 先将大系统划分成几个子系统, 再

分别建模. 对于一个复杂系统尤其类似于 JB 系统这样一类大的系统, 不可能一次性把问题域中的每一个细节都分析设计完毕, 事实上只要问题域中最基本的设计已完成就可开始进行演化过程. 至于如何来衡量, 一般来说只要问题域已明确定义、最基本的问题关键已经把握并有其解决的方法即可. 当然, 对于 JB 系统有其特殊性, 从 JB 系统到 JB、再到 JB 系统的开发, 本身就是一个不断地发展和完善的过程, 是在前期攻关的工作基础上进行的, 这和从头开发一个新系统有所不同.

4 结束语

面向对象分析建模过程是一件十分复杂的工作, 本

文所涉及的基于面向对象技术的分析建模是在开发青鸟系统实践中采用的策略及经验总结. 目前流行的各种面向对象分析建模方法各具特色, 其中涉及的相关问题也很多, 本文讨论的只是其中的一些, 目的只是希望能对软件开发者在应用面向对象技术时有所启发. 笔者所描述的基于面向对象技术的分析建模策略并不能说是很成熟的, 同时由于领域需求的各种不同的形式和细节, 软件开发者在自己的实践中发现有一些原则并不适用也是正常的. 事实上, 面向对象的分析建模过程不仅仅是依赖于面向对象技术的发展以及分析建模人员的实践和经验, 同时还依赖于领域工程、软件构架、软件过程、人工智能等相关领域的技术发展.

参考文献:

- [1] Booch G. Object-Oriented Development[J]. IEEE Trans on Soft Eng, 1986, 12(6): 117 - 122.
- [2] Coad P, Yourdon E. Object-Oriented Analysis[M]. New Jersey: Yourdon Press, 1989.
- [3] Rumbaugh J, Blaha M, Premerlani W, et al. Object-Oriented System Modeling and Design[M]. London: Prentice-Hall, 1991.
- [4] Shlaer S, Mellor S. Object Lifecycles: Modeling the World in States[M]. New Jersey: Yourdon Press, 1992.
- [5] Lubars M, Potts C, Richter C. Developing Initial OOA Models[J]. IEEE Trans on Soft Eng, 1993, 19(2): 222 - 226.
- [5] 朱三元, 钱乐秋, 宿为民. 软件工程技术概论[M]. 北京: 科学出版社, 2002.

The study of object-oriented analysis modeling policies and interrelated problems

JIA Jiong¹, QIAN Le-qiu²

(1. Institute of Computer Science, Zhejiang Normal University, Jinhua Zhejiang 321004, China; 2. Laboratory of Software Engineering, Fudan University, Shanghai 200433, China)

Abstract: The importance of object-oriented analysis modeling was discussed, some object-oriented analysis approaches were studied and compared. An object-oriented analysis approach that based on object-oriented technology in developing a series of tools of Jade Bird systems was described, and some interrelated problems were discussed.

Key words: object-oriented; object-oriented analysis modeling; subject; Jade Bird (JB) System

(责任编辑 陶立方)