

RUP 和 UML 在软件体系结构建模中的应用研究^{*}

魏艳铭, 张广泉

(重庆师范大学 数学与计算机科学学院, 重庆 400047)

摘 要 把 UML 与 RUP 作为软件体系结构建模的支持, 以某公司销售管理系统为例, 根据 UML 与 RUP 的通用建模过程逐步构建了销售管理系统的需求模型、用例动态模型、对象结构模型和实现模型。

关键词 RUP ;UML 软件体系结构 建模

中图分类号 :TP311

文献标识码 :A

文章编号 :1672-6693(2006)03-0054-05

Application of RUP and UML in Modeling Software Architecture

WEI Yan-ming , ZHANG Guang-quan

(College of Mathematics and Computer Science , Chongqing Normal University , Chongqing 400047 , China)

Abstract :UML and RUP are used to support modeling software architecture. In this paper ,based on the general modeling process of UML and RUP ,several kinds of models have been built in company sale management system ,those are requirement model、use case action model、object structure model and implement model.

Key words :RUP ;UML ;software architecture ;modeling

从 1968 年“ 软件工程 ”这个名词被提出到现在已经 30 多年了, 软件工程作为一个独立的学科较开始时已有了长足进步, 但随着进一步研究, 发现现在软件工程面临的最大问题是缺乏联系用户需要的特性和软件工程师要建造的系统特性的中间抽象^[1], 那么在设计软件时能不能像建筑行业上那样先设计一个整体的系统结构来填补这种鸿沟的抽象呢, 通过对软件体系结构的研究表明, 软件体系结构不但是一组填补鸿沟的抽象, 而且特别地它可以作为准确进行成本预算和风险分析的基础, 因而能够极大地减少系统失败的可能性。因此对于软件体系结构的设计就变为整个设计过程的关键。

如何对软件体系结构进行建模, 即如何用适当的方法把软件的体系结构模型详细精确地描述出来, 是在软件体系结构设计中一个非常重要的方面。对软件体系结构的建模不仅影响着开发人员和设计人员之间的交流和对系统的理解, 更影响到后续的工作, 因而软件体系结构的建模问题成了软件体系结构研究领域的一个核心问题。根据软件体系结构的定义^[1], 要描述一个软件的体系结构, 就是要描述

用哪些种类的部件、部件之间的连接器和软件的总结构, 它不仅要指出部件和部件的简单关系, 还必须指出它们是如何交互的, 因而根据对软件体系结构建模的侧重点不同, 对所要描述的模型的分类也不同, 本文采用 Garlan D 等提出的 5 种模型定义, 即结构模型、框架模型、动态模型、过程模型、功能模型。

1 UML、RUP 与体系结构建模

UML^[2,3]作为目前应用最为广泛的面向对象软件建模语言, 其完整、丰富的图形和符号为表达面向对象系统模型提供了有力支持, 它提供的用例图、静态图、行为图、交互图、实现图等 5 类视图不仅从多个视角描述系统的不同侧面, 而且基本上涵盖了软件体系结构模型的全部内容。如 UML 结构元素中的类、构件、节点、用例和组织元素中的包等相当于体系结构中的部件; 可以用 UML 的静态图描述部件和部件与部件之间的关系; 可以用交互图描述部件之间是如何交互的等等, 这一切都表明 UML 对体系结构建模有较好的支持。

而 UML 本身只是一种描述语言, 而不是一种开

* 收稿日期 2005-12-15

资助项目 重庆市教委科学技术研究项目(No. 040803)

作者简介 魏艳铭(1981-) , 女, 河南濮阳人, 硕士研究生, 研究方向为软件体系结构、形式化方法等。

发方法,使用 UML 快速设计和表达系统模型,必须有有效的设计方法支持。RUP^[4](Rational 统一过程模型)是 Grady Booch 等人在 Rational 公司支持下提出的一种面向对象的软件开发过程模型,它沿用原型法的思想,采用螺旋上升式的开发模式,渐增、循环、重复地完成一个软件系统的开发。RUP 在迭代的开发过程、需求管理、基于构件的体系结构、可视化软件建模、验证软件质量及控制软件变更等方面,针对所有关键的开发活动为每个开发成员提供了必要的准则、模板和工具指导。它建立了简洁和清晰的过程结构,为开发过程提供较大的通用性。

RUP 和 UML 可以作为在对系统建模时参考的主要标准和使用的主要工具。基于 UML 和 RUP 的通用建模过程可以定义为:1)需求描述——建立用户需求和功能模块;2)系统分析——确定系统中的角色、用例以及体系结构风格的选定;3)系统设计——扩展和细化各个模型,包括结构设计和详细设计;4)系统实现——确定系统实现的结构和环境模型。

在统一过程中,系统的体系结构体现在它的各种模型视图中,本文根据 RUP 和 UML 的通用过程对某小型公司销售管理系统逐步构建了需求模型、用例的动态模型、结构模型和实现模型。

2 系统建模

2.1 需求描述

销售管理系统的需求分析大致说明如下:主要包括用户登录、商品管理、业务查询与统计、系统设置4大功能子系统。用户登录子系统允许用户登录、注销和退出系统。用户可以用两种身份登录本系统:普通操作员和管理人员,不同身份登录的用户有不同的使用权限。商品管理子系统能够协助处理进货、销售货物、退货\返修、货物盘点等任务。商品管理子系统是整个系统的核心,是所有数据的来源。业务查询与统计子系统不仅可以进行库存查询、销售查询、历史记录查询、退货\返修查询,还能够对在一定时间范围内的销售情况进行统计。其中历史记录查询功能模块是在本系统中的进货、销售货物、退货\返修、盘点等的任何一个操作都储存在数据库中,能够保证查询任意一条操作记录。系统设置可以对供货商、操作人员、客户信息进行设置,如在供货商设置中,用户可以输入详细的供货商信息,包括联系方法、供货商名称和主要经营项目等信息,以便查询和维护^[5]。

2.2 系统分析

(1)需求模型。在此系统中,采用 UML 的用例图支持系统中的需求分析来建需求模型,用来描述人与系统的相互作用。从上面的需求描述里,可以看出此系统的角色可以分为:系统操作员、系统管理员、供货商、客户,系统提供的用例如下:进货、删除货物、销售货物、货物盘点、库存查询、销售查询、商品历史记录查询、退货和返修查询、销售统计、供货商信息维护、添加操作人员、客户信息维护,由此建立系统用例图如图1所示。其中管理员是操作员的泛化,实箭头表示依赖,空心的菱形箭头表示聚合,代表一种整体与部分的关系。

为了更好地发现业务过程,捕获业务过程的步骤,更好地理解系统的需求和为下一阶段域分析做准备,对系统的非功能性需求,比如系统的进货活动流程不能在用例图中表达出来,只能使用活动图来表示这种动态属性,在销售管理系统中进货功能模块的主要事件流程为:1)管理员查看销售记录并进行统计,判断商品的销售状况;2)管理员去查看商品库存情况;3)缺货则通知操作员缺货商品清单;4)操作员领取清单后联系供货商;5)供货商按缺货清单提供货物;6)操作员接收货物;7)操作员更新数据库,并完成整个操作。其活动图如图2,限于篇幅有限,本文仅详细给出进货功能模块的活动图。

(2)体系结构风格。软件体系结构设计关键是使用已有的组织结构模型即体系结构的风格,在软件开发的理论和实践中,人们设计使用多种表达软件体系结构的描述方式,形成了描述软件设计的一些规范。在分析系统需求,得出系统需求模型后,必须选择合适的软件体系结构模式即软件体系结构风格来指导设计系统的逻辑模型。本系统采用的是现在流行的 MVC(Model-View-Controller)模式,分别用表示层、应用服务层、数据层3层结构去对应 MVC 里的视图、控制器和模型的作用。表示层主要实现系统的用户接口功能,处理用户与软件之间的交互,为用户提供信息,将用户的指令进行翻译并传送给应用服务层和数据层,根据部署的需要用到了动态服务器页面、Web 控件和 Web 客户端小程序实现。应用服务层是实现系统业务逻辑的核心部分,它包括输入与存储数据的计算,验证表示层传递过来的数据,有时可根据需要采用 Web Server 技术对接口进行封装,向外提供动态服务。数据层包括了系统数据模型和数据,以及在数据库管理系统下运行的存储过程和触发器。另外将系统在3层结构划分的基础上,对应用服务层又进行了进一步划分,应用服务层总体上划分为5层:业务外观层、业务规则层、

实体控制层、数据实体层、通用系统框架^[6]。

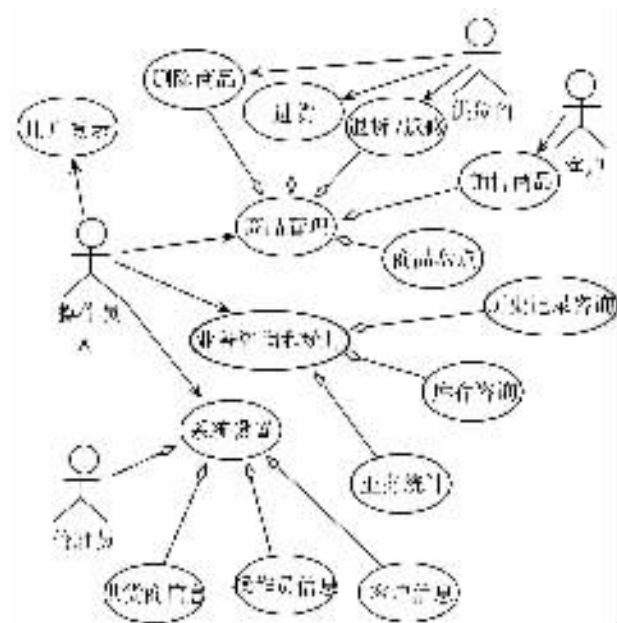


图 1 销售系统用例图

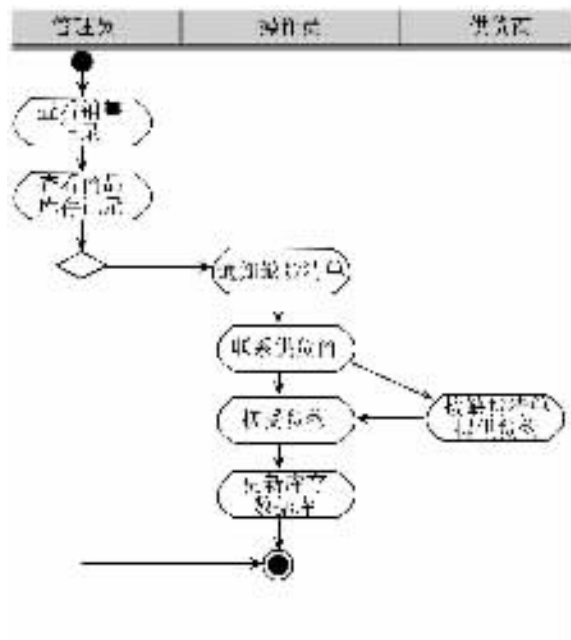


图2 进货的活动图

(3) 用例动态模型。在完成了系统的体系结构风格设计后,设计阶段的任务是以需求建立的模型为基础,更深入地考虑系统的功能目标、技术约束,扩展和细化各个模型。将用例分解为5部分进行设计,分别对应了5种不同的对象。5种对象分别为:系统界面对象、业务外观对象、业务规则对象、实体控制对象和实体对象,5种对象分别完成了体系结构风格中定义各个层次的功能,按照软件结构模型中的执行机制执行事件。

在面向对象的系统中,描述对象和对象之间的

交互情况是很有必要的,可以使用 UML 中的交互图定义出主要对象和对象之间的消息传递模型,交互图又包括时序图和协作图,一般使用时序图进行表达,因为时序图具有时间顺序,表达清晰。而 UML 提供的协作图是另一种用来表达对象交互模型的图,它集中地反映了对象之间所有的消息,尤其是集中地反映了某个对象与外界的所有消息连接。时序图与协作图是可以相互转换的,在实际中可以根据需要选用这两种图。

通过系统中定义的用例进行对象交互情景分析,可以找出系统的主要对象,并描述出对象之间的消息传递模型。在此仅列出进货用例的时序图如图3所示。

2.3 系统设计

(1)对象结构模型的类图。在完成了用例中对对象初步分析的基础上,可以对时序图中涉及的对象进行详细设计,定义出对象对应的类,根据对象间的消息定义出对象的操作。进货用例的顺序图进行详细类设计后的结果如图4所示。

在将时序图中每个对象都映射到类后,通过对对象间消息的分析和定义可以得到类的属性、方法和事件,使之成为一个完整的程序设计单位。然后就可以绘制出类图描绘系统需要实现的类,以及类间的关系。类图描述了系统中的类及相互之间的各种关系,其本质反映了系统中包含的各种对象的类型以及对象间的静态关系,另外,类图也描述了类的属性和行为以及对模型中各种成分的约束。商品管理用例涉及到的类图如图5所示。

(2)包图。利用 UML 进行系统的高层设计,包括定义包(子系统)及它们之间的关系和通信机制。由于软件系统要求符合高内聚低耦合的原则,在销售管理系统中,笔者通过实施 3 层结构分离了界面和数据,以及应用和数据的关系,定义了 4 个包及其关系,见图 6,图中的虚线表示包之间有依赖关系。把功能类似的类归入同一个包中:用户接口包处理用户输入和输出,各种用户界面窗体类和组件,作为和应用层的接口,事务包中包含的是系统所有的事务处理、业务外观、业务规则组件、控制组件等,如果管理过程中需要添加某事务,只需在本包中添加相应的类即可,数据库包提供数据库和应用层的接口以及可以永久存储的数据实体;应用包里有为其它包提供服务的类和组件如特殊的报表。

2.4 系统实现

在这个阶段中系统的实现模型包括构件图和配置图,它们表示系统实现的一些特性。其中,构件图

表示代码本身的逻辑结构,构件是系统中遵从一组接口且提供其实现的物理的、可替换的部分,构件图包含了模型代码库、可执行文件、运行库和其他构件的信息。在销售管理系统的构件图如图 7 所示,各构件之间相互通信,下层为上层提供服务,虚箭线表示依赖关系。accessdb 组件提供数据访问服务,执行 API 的请求,直接与数据库进行交互;tranc 组件提供数据转化服务,对信息服务的逻辑请求转换为数据兼容的语言(SQL 语句);func 组件提供传统的功能服务;User Interface 是用户界面接口。配置图

表示系统运行时刻的结构,利用 UML^[7]的配置图,可以从更抽象的系统设计角度上,考察每一个软件模块、每一个软件的可执行体在物理节点之间的通信方式,或者用箭头符号表示出不同节点上模块之间的通信方式。本系统采用客户端/服务器(C/S)3 层体系结构,其配置图如图 8 所示。图中的立方体表示系统中配置的节点,包括服务器和客户端,结点之间的连线表示系统之间进行交互的通信协议名称。应用服务器、数据库和客户端都存在通信路径,节点之间采用的通信协议都是 TCP/IP。

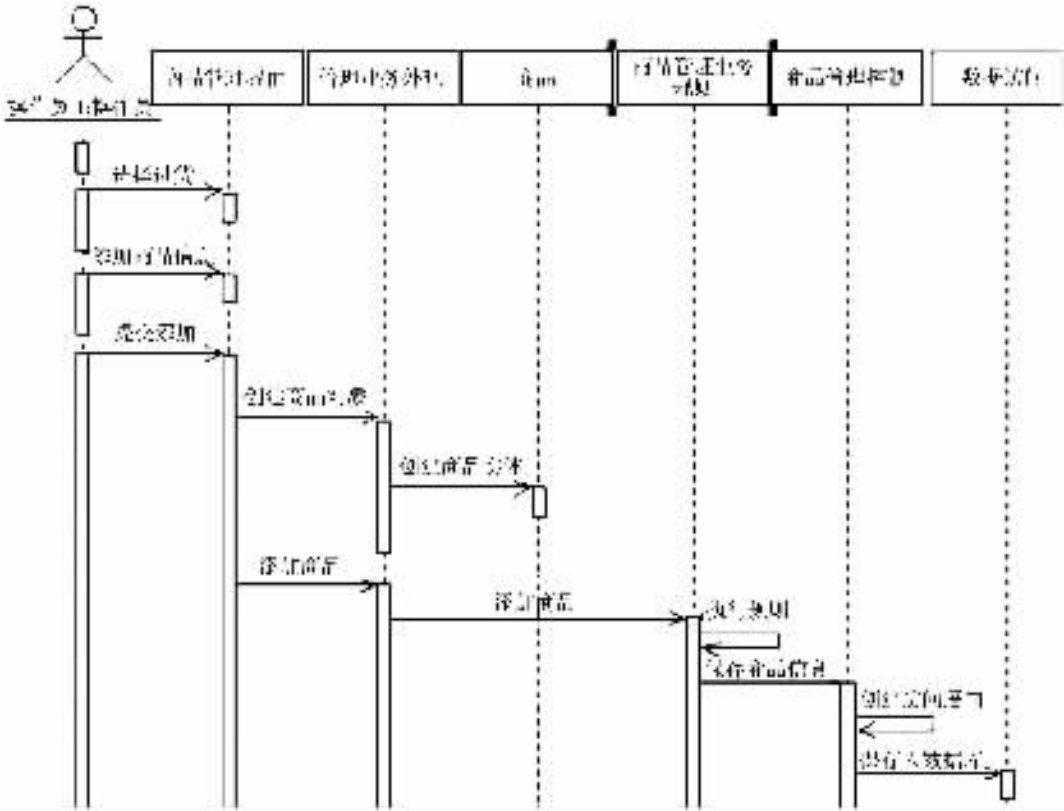


图 3 进货的时序图

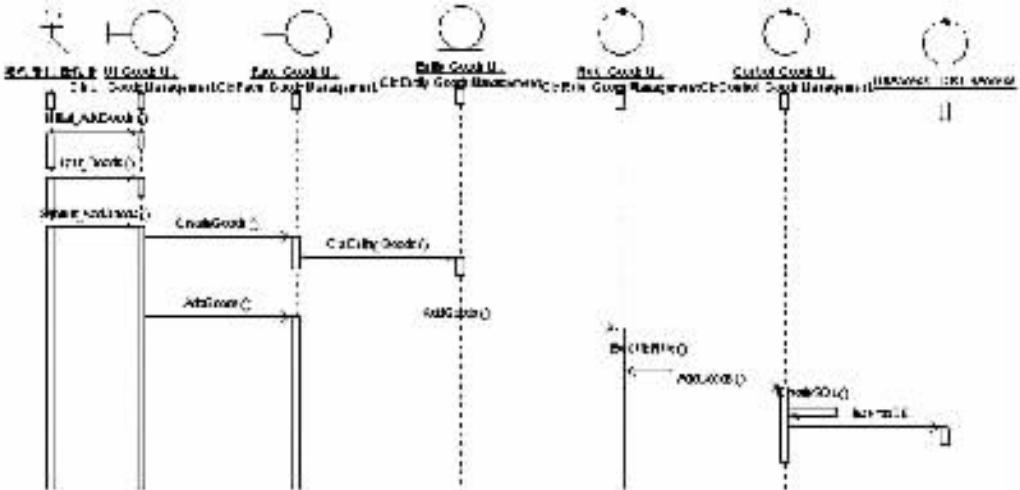


图 4 进货用例详细设计后的时序图

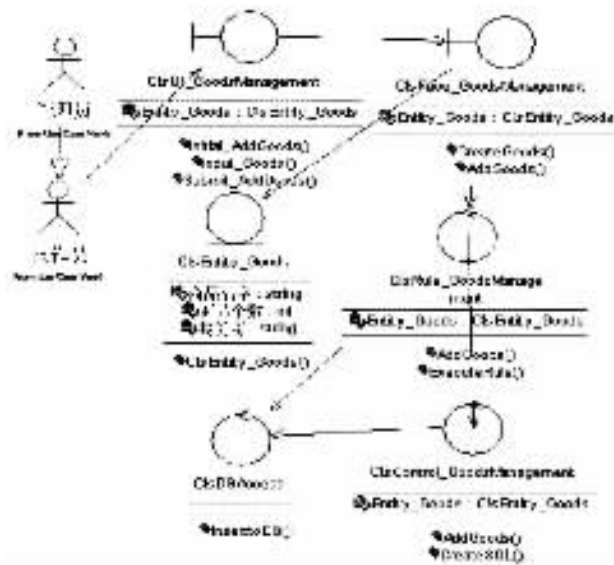


图 5 商品管理类图

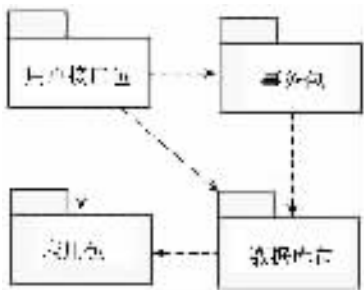


图 6 系统包图

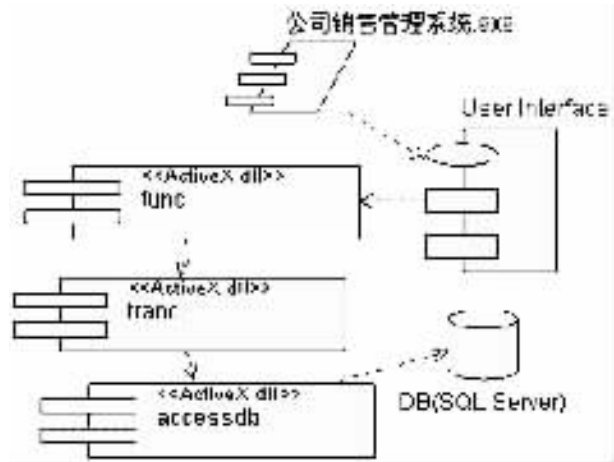


图 7 公司销售管理系统构件图

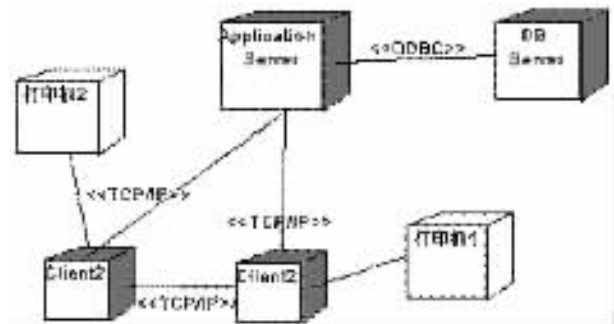


图 8 公司销售管理系统配置图

3 小结

软件体系结构设计中一个非常重要的问题是如何对软件体系结构进行建模,根据建模的角度不同,产生的模型也不一样,用什么工具来描述这些模型呢?描述这些模型时要遵循什么样的方法来指导建立稳定、可靠的软件体系结构呢?本文把 UML 与 RUP 两者结合用于软件体系结构建模中,其中 UML 作为支持建模的工具,RUP 作为建模中的方法指导,并根据两者结合的通用过程对公司销售管理系统体系结构的各个模型进行了逐步构建。UML 提供了丰富的视图从多个视角描述系统的不同侧面,可以有效运用于软件系统的建模、分析与设计。但是,作为一种通用的语言,UML 对软件体系结构的可构造性建模能力较弱,缺乏形式化语义,对体系结构的描述只能到达非形式化的层次。笔者下一步工作就是要探讨运用一种软件体系结构描述语言(Architecture Discription Languages,简称 ADL)来对软件体系结构进行建模,并对其作进一步的求精、验证工作。

参考文献:

- [1] SHAW M, GARLAN D. Software Architecture Perspectives on an Emerging Discipline[M]. New Jersey:Prentic Hall, 1996.
- [2] BOOCH G. UML 用户指南[M]. 邵维忠译. 北京:机械工业出版社 2005.
- [3] 余丽平. 软件体系结构及实例分析[M]. 北京:科学出版社 2004.
- [4] (美)KRUCHTEN P. Rational 统一过程引论[M]. 北京:中国电力出版社 2003.
- [5] 吴建,郑潮,汪杰. UML 基础与 Rose 建模案例[M]. 北京:人民邮电出版社 2004.
- [6] 常晓,王凤霞,张景,等. 基于 RUP 的 UML 建模方法研究[J]. 计算机工程与应用 2004 21:126-130.
- [7] 张广泉,刘艳. 基于 UML 的图书管理体系结构模型及实现[J]. 重庆师范大学学报(自然科学版) 2005 22(2): 1-5.

(责任编辑 游中胜)