

基于 UML 的快递系统建模

武防震, 姚国祥

(暨南大学 网络中心 广东 广州 510632)

摘 要 描述了一个用 UML 作为建模语言的快递系统的模型, 分析了快递系统的需求。对基于 UML 的快递系统建模进行了探讨。UML 具有定义良好、可视化、可扩展和功能强大等优点, 成为面向对象建模的事实标准。

关键词 统一建模语言; 快递系统; 建模; 统一软件开发过程

中图分类号: TP399

文献标识码: A

文章编号: 1005-3751(2003)03-0012-03

Express System Modeling Based on UML

WU Fang-zhen, YAO Guo-xiang

(Network Center Jinan University, Guangzhou GD 510632, China)

Abstract Describes a model of the express system with UML as modeling language. Analyzes the requirement of the express system. Discussed the UML-based express system modeling. UML become the factual standard of OO modeling on account of its good definition, visualization, expansive and strong functions.

Key words UML; express system; modeling; RUP

1 引 言

自从 1997 年底统一建模语言(UML)成为对象管理组织(OMG)推荐的建模语言以来,已经得到了广泛的应用^[1]。UML 成为 OO 建模的标准语言,获得了业界的广泛支持。快递软件系统在确定需求以及开发过程中,需要意义确切、表示明快的模型来和用户交互,故可采用 UML 作为建模语言。

2 UML 模型元素概述

UML 的主要内容包括五种视图模型和十类图形,视图模型借助于各类图形从不同视角(侧面)对所建模系统进行描述。UML 可以满足从需求分析到系统实施的整个过程中的建模需求,其中用例视图模型、静态视图模型和实现视图模型属于静态建模,行为视图模型和交互视图模型属于动态建模^[2~4]。

(1)用例视图模型(Use Case View)。从系统用户的角度描述系统的功能集合。可用图形主要为用例图(Use Case Diagram),用例图中使用角色表示用户,使用用例表示系统的某一功能。

(2)静态视图模型(Static View)。描述系统各部分的静态结构。涉及图形包括类图(Class Diagram)、对象图(Object Diagram)和包图(Package Diagram)。类图用于表示系统中的类及类之间的联系。对象图是类图的具体对

象实例。包图用于描述系统的分层结构,由包或类组成。

(3)行为视图模型(Behavior View)。描述系统的动态模型和组成对象间的交互关系。包括状态图(State chart Diagram)和活动图(Activity Diagram)。状态图表示类的对象可以呈现的状态和这个对象从一种状态到另一种状态的转换。活动图描述为满足用例要求所要进行的活动以及活动间的约束关系。

(4)交互视图模型(Interaction View)。描述对象间的交互关系。包括时序图(Sequence Diagram)和协作图(Collaboration Diagram)。时序图强调对象之间随时间变化的交互过程,而协作图强调对象之间在空间上的交互关系。

(5)实现视图模型(Implementation View)。描述系统最终实施时的软件和硬件结构。包括构件图(Component Diagram)和配置图(Deployment Diagram)。构件图描述系统的软件部件结构,配置图描述系统的硬件拓扑结构和构件在网络上的分布。

3 用 UML 建模快递系统

从应用的角度看,对一个应用系统的建模,首先是通过用例模型描述需求,其次是根据需求建立系统的静态模型、行为模型和交互模型,最后建立系统的实现模型。

3.1 需求建模

UML 的可视化用例图形用于担当确定系统的确切需求这一工作,可以方便地和用户通信、交互,其中主要工作在于确定系统的主要用例。快递企业的主要业务就是做

为寄件客户(寄出快件客户) 和派件客户(接受快件客户) 之间的一个中介 , 如图 1 所示。在快递企业内部 , 必须完成将快件从寄件客户所在站点运输到收件客户所在站点 , 此外 , 还必须将从寄件一开始就产生的一些信息如收件人、收件地址、费用等 , 传达到客户和公司内部的业务、财务等相关人员手中。由此看出 , 在快递企业里需要保证两条流的畅通 , 一条是实际快件运输流 , 另一条是快件相关信息传播流。在传统快递企业里 , 快件相关信息的流动是通过类似于快件的单据的运输来进行的。快递软件系统将快件的相关信息提取到电脑里 , 通过电脑网络传播 , 保证信息流的畅通 , 这样做省去了大部分单据 , 降低了成本。



图 1 快递服务示意

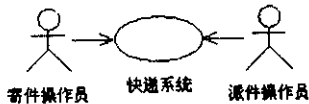


图 2 实际快递系统示意

从上述对快递系统的分析看出 , 快递系统的用户除了企业外部客户外 , 还包括企业内部的业务、财务和管理等人员。快递系统需提供的主要功能除了面向企业外部客户的快递服务外 , 还应该面向企业内部的财务管理、客户管理和人事管理等。但是 , 在实际运作时 , 快递企业还不能提供像银行的 ATM 一类的服务 , 即和电脑系统进行交互的并不是最终客户 , 而是快递企业的电脑操作人员 , 所以将图 1 模型改为用图 2 表示更合适 , 但为了讨论方便 , 本文采用了图 1 形式。另外这里的“ 快递服务 ” 概念太大 , 可以分拆为下单、运务管理和查询三个部分。快递业务过程中的主要事件流为 : ① 寄件客户下单 ; ② 业务人员到寄件客户处取得快件后 , 将快件入库 ; ③ 快件被运输到目的地 ; ④ 业务人员将快件出库后送到派件客户手中。由此得到的一个简化快递系统的业务部分用例视图如图 3 所示 , 其中快件在企业内部的中转流通称为运务管理。

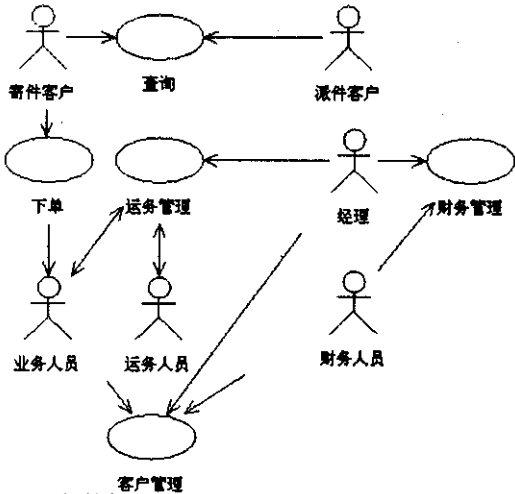


图 3 快递系统用例视图

需说明的是 , 对系统的非功能性需求 , 如用例图中某个用例的内部活动过程等 , 不能在用例图上进行描述 , 故还用活动图等动态建模图形对内部过程加以描述。不过这里用活动图等进行的只是概念层的描述 , 和后面分析设计阶段的说明层描述及编码阶段的实现层描述不同。这里以用例“ 运务管理 ” 为用例活动图加以说明 , 见图 4。

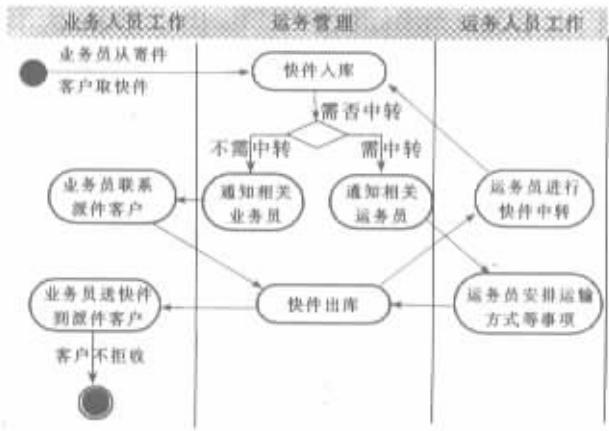


图 4 运务管理活动图

3.2 静态模型

组成系统的主要类构成了系统的静态结构 , 因此这里的主要工作在于确定系统里的主要类 , 并用类图描述它们。快递系统里涉及的主要类有快件类、客户类、员工类和站点类 , 其中员工类是企业内部各类员工的超类 , 站点类用来描述分布在各地的站点 , 这样得到的简化类图如图 5 所示。在编程实现时 , 如果认为“ 快件 ” 类粒度过大 , 可将其划分为几个小类。

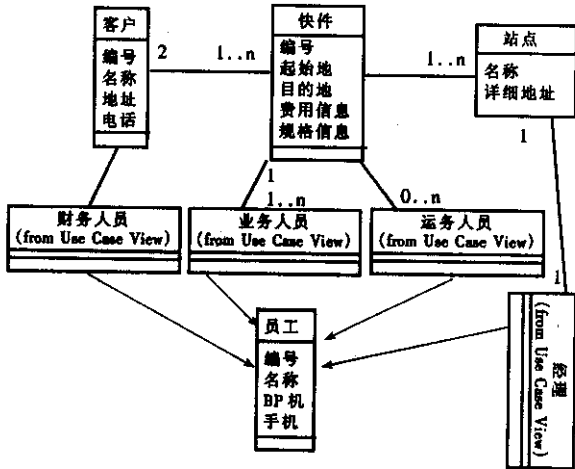


图 5 简化类图

3.3 行为模型

对象的行为用状态图和活动图来描述。状态图适合描述跨多个用例的单个对象的行为 , 活动图用于展现多个对象活动的总次序 , 可根据具体对象的需要选用图形。这里用状态图对“ 快件 ” 类的对象行为进行了分析 , 见图 6。

3.4 交互模型

消息是发送对象要求目标对象执行某个操作的一个

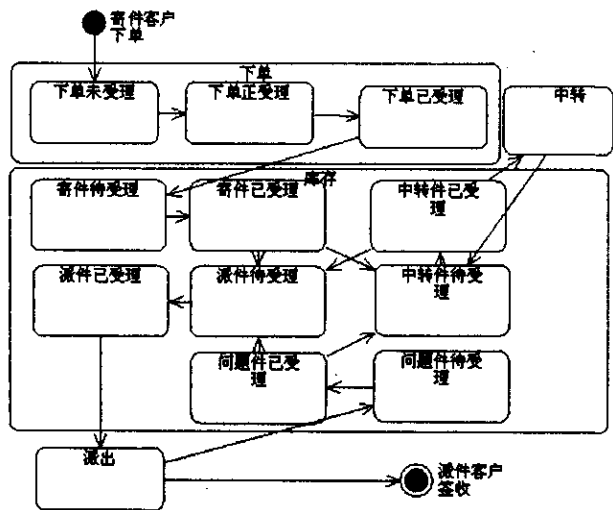


图 6 对象状态图

请求,对象间的交互主要是通过对象间消息的发送来完成的。时序图中对象的交互表现为对象之间随时间的消息发送过程,而协作图中的对象的交互表现为两个对象相互发送的所有消息。时序图和协作图所记录的内容基本上是相同的,且两者可以相互转换。图 7 用时序图描述了下单管理中的对象交互。

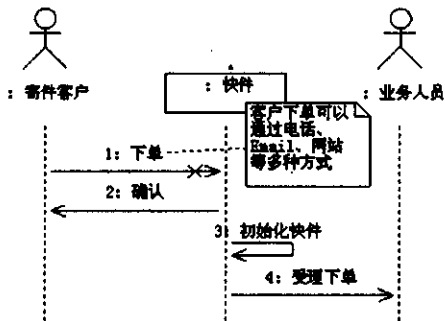


图 7 交互模型时序图

3.5 实现模型

构件图中的构件可能是源代码文件、二进制文件和可执行程序单位,用于对系统实施时的软件逻辑结构进行描述。配置图中的主要元素有处理器、设备、进程和处理器与设备之间的连接。构件图在系统实施时才使用,这里只给出一种简单的快递系统的配置图,见图 8。

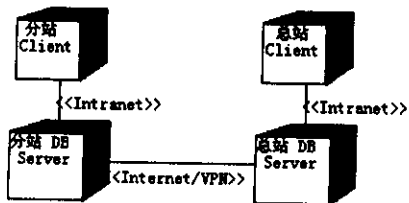


图 8 快递系统配置图

4 UML 与软件开发过程的结合

一种软件开发方法包括建模语言和对开发过程的规范化定义两部分,而 UML 只是一门 OO 建模语言,其中

不涉及软件过程概念,故用 OO 方法开发快递系统必须将 UML 和一种开发过程结合起来进行。

UML 作为一门标准化的建模语言,支持常用的开发过程,如 RUP、OPEN 和 OOSP 过程等^[3]。由于 Rational 公司的统一过程(RUP)是 UML 的三位作者在提出 UML 的基础上推出的,结合了一些 OO 开发的已有经验,故更适合和 UML 结合,从而使统一建模语言和统一软件过程相统一组成一种完整的软件开发方法。RUP 的核心思想认为软件开发是一个迭代、递增式的开发过程,是一个用例驱动的开发过程,是一个以体系结构为中心的开发过程。以此为基础,RUP 把对一个系统的开发周期按时间和核心 workflow 划分为一个二维空间。在时间维上从组织管理的角度划分为四个阶段,即初始阶段、细化阶段、构造阶段和移交阶段,每个阶段以相应目标(称为里程碑)的完成为结束。这四个阶段又划分为各自相对独立的子阶段,通过子阶段的实施,最终达到规范地、有步骤地完成整个系统的开发。核心 workflow 从过程静态的一面描述,将每个阶段要完成的任务划分成一些 workflow。

5 支持 UML 的 CASE 工具

实际使用 UML 时需要 CASE 工具的帮助。目前支持 UML 的 CASE 工具有 RationalSoft 的 Rose、Microsoft 的 Visio 以及一些大型开发环境(如 MS Visual Studio、Borland Jbuilder 等)中自带的专用建模工具。CASE 工具(以 Rose 为例)除了可以方便地构造 UML 的模型元素并检查所建模型是否符合 UML 的规则外,还可以自动生成程序的框架代码,以及对已有代码生成其模型,从而在开发过程中使模型和代码保持一致^[4]。本文中所用 UML 图形是在 Rose 中构造的。

6 结束语

在开发快递系统时使用 UML 建模,利用了 UML 的优点,但同时也会遭遇 UML 的一些不尽如人意的地方,如建模元素太复杂、不可以形式化推导、一些建模元素如活动图用途还不清晰等。和任何新生事物一样,UML 也在发展,新版本的 UML 2.0 计划 2002 年底推出,相信 UML 会更好地满足建模需要。

参考文献:

- [1] OMG. UML tutorial[EB/OL]. <http://www.omg.org/uml/>, 2002.
- [2] 刘超,张莉.可视化面向对象建模技术[M].北京:北京航空航天大学出版社,1999.
- [3] 蒋慧,吴礼发,陈卫卫,等. UML 设计核心技术[M].北京:北京希望电子出版社,2001.
- [4] Boggs W, Boggs M. Mastering UML with Rational Rose[M]. 邱仲潘译.北京:电子工业出版社,2000.