

# OOSDM 的对象模型与系统构造\*

王立福 阮 闯 杨美清

(北京大学计算机科学与技术系, 北京 100871)

**摘要** 随着面向对象实现技术的快速发展,面向对象的各种软件开发方法正在相互融合,并进入实施应用阶段,从而有力地促使了这一方法逐步走向成熟.基于“青鸟工程”和现有的面向对象方法,我们提出了 OOSDM(Object-Oriented Software Developing Method),本文主要介绍 OOSDM 的对象模型——结合事件对象,以及如何以这一对象模型进行软件系统构造.

**关键词** 对象模型,需求模式,系统模型,系统构造.

## 1 OOSDM 概述

国家“八五”重点攻关项目“软件工程开发环境(CASE)的标准化与实用化”(简称“青鸟工程”)中的一个主要的子专题就是面向对象软件开发规范的制定及相应支持工具的开发.

结合“青鸟工程”的实践,综合现有面向对象软件开发方法,我们给出了一套面向对象软件开发的观念、模型和过程,简称为 OOSDM(Object-Oriented Software Developing Method).其中吸收了现有典型面向对象软件开发方法的优点,如文献[1—3]中对于类及对象及其关系的符号表,文献[4]中将系统模型分为结构、行为和功能,文献[5]中的需求模式(Use Case)、对象交互图(Object Interaction Diagram)以及采用多种对象划分系统模型的思想等.

OOSDM 的特点是系统性、集成性和可操作性. OOSDM 的系统性体现在它覆盖了软件开发的整个过程,对于每个阶段均提供了实施步骤和解决方案. OOSDM 的集成性体现在其提出的结合事物的对象概念,它使软件开发各个阶段的表达有机地联系起来,而且能够表达框架(architecture)<sup>[6]</sup>和软件部件(Componentware)<sup>[7]</sup>中的对象. OOSDM 的可操作性除了体现在它的系统性外,还体现在获取用户需求的系统化方法、开发增量的选择、风险的判断等方法. OOSDM 之所以能具有这些性质是因为采用了需求模式概念来表达系统外部行为和功能;同时提出了结合事件的对象概念,弥补了通常面向对象方法中对象表达能力的不足;并将系统需求与系统模型自由地结合起来.

本文主要介绍 OOSDM 的对象模型和软件系统构造.

\* 本文 1994-06-18 收到,1994-12-18 定稿

作者王立福,1945年生,副教授,主要研究领域为软件工程.阮闯,1967年生,博士研究生,主要研究领域为面向对象方法,软件开发平台.杨美清,女,1932年生,中国科学院院士,教授,主要研究领域为软件工程.

本文通讯联系人:王立福,北京 100871,北京大学计算机科学与技术系 CASE 研究室

## 2 OOSDM 有关系统建模的基本概念

除通常面向对象方法的一般概念外,OOSDM 的核心概念有两个:需求模式(Use Case)和结合事件的对象(Object With Event,简称 OWE 对象).需求模式及其相关概念来自 Objectory<sup>[5]</sup>,用以表达系统的需求,在使用的形式上有所调整.

本文提出了结合事件对象(OWE)的概念,用以增强现有对象概念的表达能力.在 OOSDM 中,我们用 OWE 及其相关概念来表示系统构成.

### 2.1 需求模式及其相关概念

需求模式(Use Case)用以描述用户与系统间的交互活动.需求模式之间的关系有两种:引用(use)和附加(expand);与需求模式相关的概念是对象交互图(Object Interaction Diagram),它以图形的方式展示出在需求模式所描述的交互过程中系统内部涉及的对象及其活动.

#### (1)需求模式(Use Case)

需求模式是用户与系统间交互活动的表达.每个需求模式都是一类用户的一种特定的使用系统完成一定功能的方式.需求模式通常包括:需求模式名称、用户类型、基本过程、附加过程、异常情况等.需求模式中过程和情况的描述采用类似脚本(Script)的结构化过程式语言,内容反映的是用户使用系统时的一系列操作和系统相应完成的功能,其中还可以包括界面描述及对其它需求模式的引用.必要时,对于一些需求模式还可以加入复杂性评价、风险评价、重要性评价等内容.

系统的每一类用户一般都有多个需求模式.全部的需求模式构成了系统的需求<sup>[5]</sup>.

#### (2)对象交互图(Object Interaction Diagram)

对象交互图(Object Interaction Diagram,简称 OID)是以图形的方式细致地展示在需求模式描述的交互过程中系统内部涉及的对象和它们的活动.每个需求模式对应一个 OID. OID 由系统边界、外部事件、对象、服务请求等部分组成.图 1 给出 OID 的符号及图示.

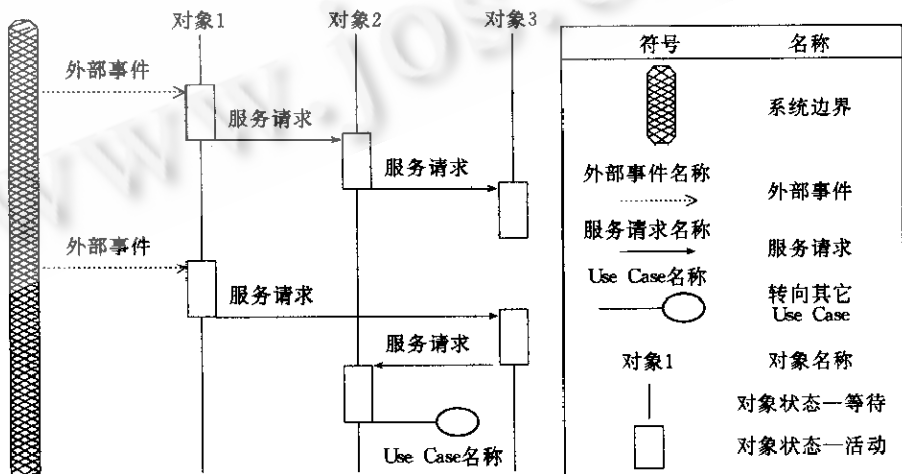


图1 对象交互图

### 2.2 结合事件的对象及其相关概念

采用面向对象方法描述系统模型的基本概念是对象. 一个对象类是一个集合, 在该集合中, 每个元素均是对象, 它们具有相同属性、相同方法(或称为服务)、与其他对象类中的对象具有相同关系、并具有相同语义. “类及对象”意指一个对象类及属于该类的所有对象.

在 OOSDM 中, 为了增强对象的表达能力, 对通常对象结构作了一些扩展, 我们把这样的对象称为结合事件的对象(Object With Event, 简称 OWE)(在本文中, 如果不作解释, 对象即指 OWE 对象), 它构成了 OOSDM 中的系统模型. 与对象相关的概念是对象实例、实例连接、“一般—特殊”结构、“整体—部分”结构、消息连接等.

OWE 对象类的 BNF 范式如下:

```

<对象类> ::= <对象类声明> <对象类定义>
<对象类声明> ::= class <类名>, { <父类> } * {
    attribute                { <属性> } *
    service                   { <方法> } *
    invariant                  { <类约束断言> } *
}
<属性> ::= <属性类型> <属性名>
<方法> ::= [ <接收事件> ] <方法名> ( <参数> ) [ { <发送事件> } * ]
<对象类定义> ::= { <方法定义> } *
<方法定义> ::= [ <接收事件> ] <类名> ::= <方法名> ( <参数> )
                                                [ { <发送事件> } * ] {
    pre-invariant             <前置断言>
    fun-dsp                   <功能描述>
    post-invariant            <后置断言>
}

```

图 2 OWE 对象类定义的 BNF 表示

方法的语义用它的前置断言和后置断言来描述, 前置断言给出了方法执行前应满足的条件, 后置断言给出了方法执行结束后满足的条件. 方法中的前置断言和后置断言并不能完全规范对象的行为, 因而引入对象类的约束断言. OWE 类及对象的其它特征与通常的类及对象相同. 例如封装、继承等. 它与通常对象的主要区别是对象间通讯除了通过“请求—服务”方式外, 还可以通过事件触发. 另一个区别是在对象模型中以断言方式给出了对象方法的语义描述.

图 3 是对象间交互示意图(其中的事件服务在实现中既可以体现为系统一级的支持, 也可以体现为一些对象).

### 3 OOSDM 的系统构造

为了更清晰地描述系统在结构、行为和功能及其演化, 从系统构成的角度, OOSDM 把对象划分为: 实体对象、连接对象、界面对象和功能对象.

**实体对象** 软件系统是一个信息处理系统, 实体对象就是用来描述系统所处理信息的模型. 实体对象的方法中包含了相应的信息处理方法. 实体对象对应于应用领域中的对象.

**连接对象** 实体对象是系统处理的核心对象, 还存在一些将各个部分连接起来的对象, 它们对系统的结构起着重要的作用. 我们把这些对象归类为连接对象. 在采用面向对象思想开发的系统中, 这种对象有时体现为系统提供的连接机制或系统服务.

**界面对象** 界面对象主要体现了系统外部行为. 用户界面对象、网络通讯对象等均可以看作界面对象. 这些对象常具有被外部事件驱动的特征.

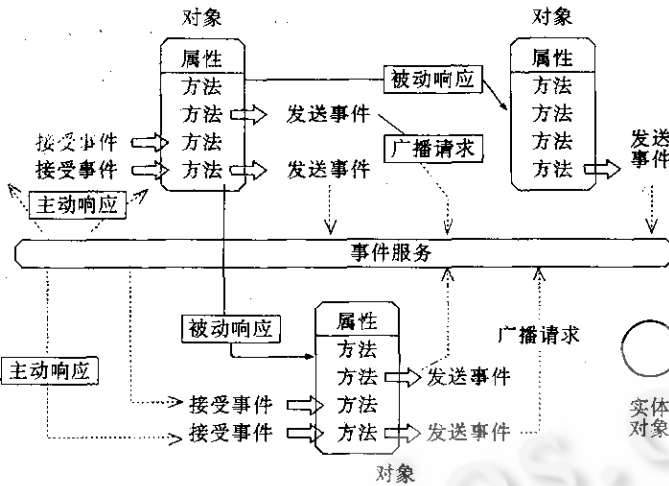


图3 对象间通讯方式

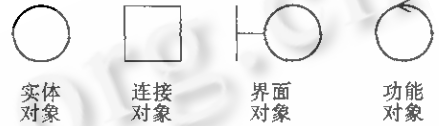


图4 四种对象的符号表示

**功能对象** 系统总是用来完成某些功能的. 这些功能有的比较简单, 只与一个对象有关, 可以用相应对象的一个或多个方法来实现. 大部分功能比较复杂, 涉及多个对象. 这些复杂功能的实现不宜放在其中的某个对象中. 采用独立的功能对象来表示可以使这些功能更加明确, 并且可以使系统在功能方面的演化更加方便. 此外, 系统中不属于上述 3 种的对象也可以划为功能对象.

在系统分析中, 一个具体的对象可能兼有几种对象的角色. 不过, 可能更侧重于表达系统的某个方面. 它们的符号表示如图 4.

### 3.1 OOSDM 的系统模型

在 OOSDM 中, 系统模型由结构模型、行为模型和功能模型 3 部分组成, 它们构成了 OOSDM 的核心模型.

(1) 结构模型 OOSDM 的系统结构模型分为“个体模型”和“群体模型”. 个体模型即是 OWE 对象的定义, 它包括属性、方法、触发事件等部分. 群体模型即是通常的类及对象关系图, 包括类及对象间的实例连接、“整体—部分”结构、“一般—特殊”结构等各种关系.

(2) 行为模型 在 OOSDM 中, 每个对象(类)采用通常的状态转换图(State Transition Diagram)来表示其整个生存周期的合理行为. 整个系统的行为采用细化的需求模式及其 OID 图形来表示, 还可以附以类及对象间的消息连接图.

(3) 功能模型 每个类及对象的方法所完成的功能可以采用伪码、断言(包括前置断言和后置断言)等方式来定义. 对于比较复杂的方法, 可以采用流程图来表示. 整个系统的功能采用细化的需求模式及其 OID 图形来表示.

### 3.2 软件系统的构造

一般来说, 系统开发主要包括需求获取、分析、设计、编码、测试、包装等几个阶段. OOSDM 的开发模型如图 5 所示.

在需求获取阶段, 主要的工作是与用户交流, 获得正确的需求, 得到各个系统的全部需求模式描述. 在分析阶段, 主要根据需求模式, 发现系统对象, 建立相应的对象交互图, 即根据系统需求得到系统的逻辑模型. 用 4 种对象划分系统结构有助于得到一个比较稳定的系

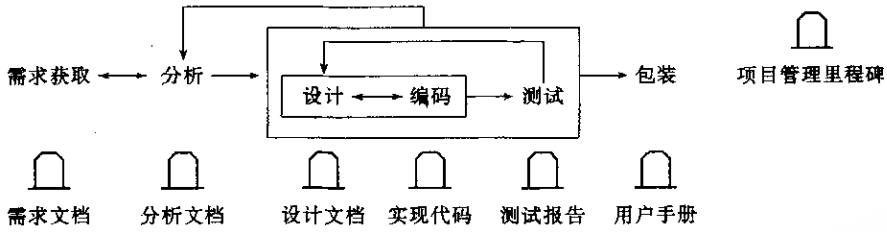


图5 OOSDM的软件开发模型

统模型表达. 在设计阶段,OOSDM 采用增量开发策略,增量的单位为一个需求模式. 对于已实现的需求模式,用需求获取中相应的需求模式描述作功能测试. 这些结果组成了测试报告. 最终,在系统包装时的用户手册也基于相应的需求模式描述.

上述面向对象软件开发模型体现了迭代和增量开发的特点. 其独特之处是,可以获得稳定的分析模型,使分析和设计之间在系统结构方面没有大的变化,过渡则比较自然,并且使设计到测试阶段具有较高的自由度.

基于应用框架的软件系统开发将成为面向对象软件开发的一种主要方式. 在 OOSDM 中,基于应用框架的系统结构如图 6 所示.

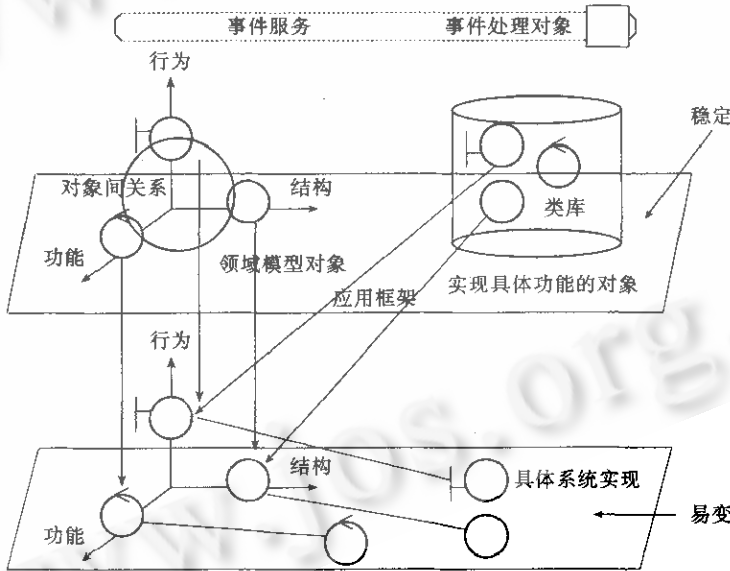


图6 基于GUI应用框架的系统构造

其中,事件服务及其相关的事件处理对象是与领域无关的一种机制. 领域对象模型通常针对一个领域进行抽象,可以为同一领域多个应用程序所复用. 领域对象模型中的实体对象、界面对象和功能对象大多是以抽象类的形式出现,可以通过继承方式作进一步的具体实现,形成该领域功能对象类库. 一般来说,应用框架是相对稳定的. 在这一框架下,可以通过“继承”或通过类库的剪裁、组装来实现一个特定需求的系统,或完成已有系统的演化.

由上可知,OOSDM 的突出特点是:用户需求与系统模型是有机联系起来的. 首先,通过一组需求模式描述用户的需求. 其次,对于每个需求模式建立对象交互图. 从需求模式到

OID 的转换过程既是系统观察角度从外部转向内部的过程,也是一个从系统的非形式描述转向形式化描述的部分.继之,通过分析参与活动的对象以及它们之间的联系,建立系统结构模型、行为模型和功能模型.如上所述,在 OOSDM 中,需求模式、OID 与系统模型紧密相关,甚至是其组成部分.在构造系统模型中,OID 的外部事件名和服务请求名均可直接对应到对象模型中的方法名称.并且,对象方法的简要功能描述可以通过相应的需求模式中的有关系统功能的描述得到.

#### 4 结束语

面向对象软件开发方法至今还是一个年轻的领域,随着软件开发技术的发展,将逐步走向成熟.OOSDM 是在新的开发技术(框架、软件部件等)的推动下,综合现有优秀软件开发方法,并结合“青鸟工程”的实践而提出的一个系统化的面向对象软件开发方法.我们将在进一步的实际工作中不断完善.

**致谢** 在本文的撰写过程中,得到了中国科学院软件研究所仲萃豪研究员的真诚帮助,在此一并表示衷心谢意.

#### 参考文献

- 1 Booch G. Object-oriented design with applications. Benjamin Cummings, Redwood City, CA, 1991.
- 2 Peter C, Edward Y. Object-oriented analysis. Prentice-Hall, Inc., (Englewood Cliffs), 1990. 中译本:邵维忠,廖钢成,李力.面向对象的分析.北京:北京大学出版社,1994.
- 3 Peter C, Edward Y. Object-oriented design. Prentice-Hall, Inc., (Englewood Cliffs), 1991. 中译本:邵维忠,廖钢成,苏渭珍.面向对象的设计.北京:北京大学出版社,1994.
- 4 Rumbaugh J, Blaha M, Premerlani W *et al.* Object-oriented modeling and design. Prentice Hall, 1991.
- 5 Jacobson I *et al.* Object-oriented software engineering—a use case driven approach. Addison-Wesley, 1992.
- 6 Vlissides J M, Linton M A. Unidraw: a framework for building domain-specific graphical editors. In: Proceedings of the ACM User Interface Software and Technologies'89 Conference, November 1989. 81—94.
- 7 Lavoie D, Baetjer H, Tulloh W *et al.*. Component software: a market perspective on the coming revolution in solutions development. Distributed Computing Monitor, January 1993, 8(1):1.

## OBJECT MODEL AND SYSTEM CONSTRUCTION IN OOSDM

Wang Lifu Ruan Chuang Yang Fuqing

(Department of Computer Science and Technology, Beijing University, Beijing 100871)

**Abstract** Object-oriented software developing method comes into a fusion period and is becoming mature along with object-oriented implementation technology is developing rapidly. Based on Jade Bird engineering and current various methods, the authors proposed a object-oriented software developing method named OOSDM (object-oriented software developing method). This paper gives its object model and system construction.

**Key words** Object model, use case, system model, system construction.