

文章编号: 1002-0411(2003)03-0225-04

# 基于面向对象机制的复杂系统的模式构造

尹春华<sup>1</sup> 顾培亮<sup>1</sup> 陈雷<sup>2</sup>

(1. 天津大学管理学院 天津 300072; 2. 辽宁大学计算中心 沈阳 110036)

**摘要:** 本文通过论述面向对象机制及面向对象机制控制复杂性的原则,把面向对象思想引入到复杂系统的分析研究中来,然后利用具有面向对象机制的 UML 对复杂系统中的某一对象进行了模式构造.\*

**关键词:** 面向对象; 复杂系统; UML

**中图分类号:** TP13 **文献标识码:** B

## MODELING CONSTRUCTION ABOUT COMPLEX SYSTEM BASED ON OBJECT-ORIENTED MECHANICS

YIN Chun-hua<sup>1</sup> GU Pei-liang<sup>1</sup> CHEN Lei<sup>2</sup>

(1. Management School of Tianjin University, Tianjin 300072; 2. Computing Center of Liaoning University, Shenyang 110036)

**Abstract:** This paper discussed the object-oriented mechanics and the rules by which object-oriented mechanics controlled the complex system. Its target was to take the idea about object-oriented into the study field of complex system. At last the UML was used to describe one object of complex system.

**Keywords:** object-oriented; complex system; UML

### 1 引言 (Introduction)

近年来,随着计算机技术、信息技术、人工智能、管理科学、决策科学、心理学、行为科学和组织理论等学科与技术的发展,针对复杂系统研究的各种技术与方法也日益丰富起来.在选择研究复杂性科学的方法时,最重要的一点是看这种方法是否能把复杂性当作复杂性来处理.经典科学相信客观世界本质上是简单的,复杂性是披在简单性之上的面纱,随着科学的发展必将揭开这层面纱,把复杂性还原为简单性.因此,在面对复杂的问题时,总是设法把复杂性简化掉,即把复杂性当作简单性处理,当对象是典型的简单系统,或者属于不够典型的复杂性问题时,这样处理是可行的或近似可行的.但当对象属于真正的复杂性问题时,这样处理必然把产生复杂性的根源简化掉,得到的结果不再能够反映对象的固有特性.因此能否把复杂性当作复杂性处理是衡量一个复杂性研究的原则之一.用面向对象的机制对复杂性问题进行处理是目前分析复杂性问题的重要方法之一.它符合上述的把复杂性当作复杂性来处理这一原则.面向对象的机制最初在计算机科

学中是为了解决系统的复杂性而提出的.面向对象的机制在面对复杂问题时,将复杂问题进行过程抽象,然后将其描绘成一个个对象,对对象进行研究和分析.因为每个对象都包含对象名称、属性、操作方法,源体的基本特征没有被简化掉,基本属性都包含其中,因此就保留了其原有对象的复杂性.利用面向对象机制研究复杂系统的另一个优点是:面向对象机制更注重系统的信息与功能,它以系统的功能为中心进行对象划分,有助于复杂系统分析的灵活性、继承性与可重用性.因此说,面向对象机制是描述复杂性的一个重要方法.

### 2 面向对象机制 (Object-oriented mechanics)

面向对象机制主要是强调人们应更专注于一个复杂系统中的重要问题.面向对象的基本机制是对象、消息、方法、类、实例及继承.

• 对象:是含数据以及对之如何操作的方法的模块.即有着特殊属性(数据)及操作方法(过程)的实体.

• 消息:是发送给一个对象的用以改变对象的状态或返回一个值的请求.

\* 收稿日期:2002-08-25  
基金项目:国家自然科学基金资助项目(70041041)

• 方法:当一条消息被发送给一个对象时,相对的进行响应回答的过程或函数方法.

• 类:是对一组几乎相同的对象的描述,由概括了一组对象共同性质的方法和数据组成.对象是类的一个实例.

• 继承:是自动共享类、子类与对象中方法与数据的机制.

综上所述,面向对象可以定义为:

面向对象 = 对象 + 分类 + 继承 + 通过消息的通讯

### 3 面向对象机制控制复杂性的原则 (The rules for controlling complexity based on object oriented mechanics)

面向对象的主要特点是能够增进复杂性控制.控制复杂性的原则既适应于 OOA( Object Oriented Analysis) 也适应于 OOD( Object Oriented Design), 主要有以下几条原则:

#### 3.1 抽象

抽象包括过程抽象和数据抽象.抽象原则即忽略问题中与当前目的无关的方面以便更充分地集中于与当前目的有关的方面.运用抽象原则时,人们认识到现实世界中的事物是复杂的,因此不是试图理解完全的事物,而是仅仅选择它的一部分.过程抽象也常说成是功能/子功能抽象.把复杂系统处理分解成一些子步骤是控制复杂性的基本方法之一.它并不是 OOD 的基本抽象形式,但它在说明和描述服务时能够发挥作用.另一个更有力的抽象机制是数据抽象.这一原则可以做为思维组织和系统责任描述的基础.它根据应用于该类对象的操作来定义数据类型.其约束是这些对象的值只能通过这些操作来修改或观察.运用数据抽象,设计者可以定义属性并有权对这些属性进行操作服务.属性及其服务可看作一个固有的整体.

#### 3.2 封装

在面向对象系统中,我们将信息与处理信息的功能组合起来,然后将其封装成对象,这称为封装(encapsulation).封装(信息隐蔽)原则在研究一个复杂系统结构时,系统的每个成分应该封装或隐蔽在一个单一的设计模块中,每个模块的界面定义尽量少地展露它的内部工作,这有利于隐藏其复杂性,减少复杂工作.如果把分析结果中最易变的部分封装起来,那么需求变化就不太可怕了.同时它把相关的内容结合在一起,减少了各部分工作之间的交叉.因

此,封装一方面是把相关的东西结合在一起,其形式之一是数据抽象,另一方面是减少接口,其形式之一是消息通信.通过消息进行通信是为接口而设的主要原则.它在封装机制中把执行活动的细节封装在消息接收者内部.

#### 3.3 继承(刻画一般性和特殊性)

继承原则形成了明确表示共同性这一很有意义的技术基础,它可以一次性地说明公共的属性和服务,同时也可以特殊情况下特化和扩充这些属性和服务.

继承机制的优点之一是可以把复杂系统简化而又不能减少造成其复杂性的根源,如图 1 所示,它把生物系统中的哺乳动物的共同属性放在父对象中,这样其他子对象的功能属性就相应减少了,降低了系统的复杂度.

继承机制的优点之二是易于改变,适应外界变化,具有灵活性.因为复杂系统是一个开放的系统,它始终处于变化中,当某个共有的属性变化时,只需更改其父辈的属性,其它子对象就会自动继承其特征属性.如图 1 所示,当所有对象变成冷血动物时,只要改变哺乳动物对象,猫、狗、人、海豚和其他子对象就会自动继承哺乳动物新的冷血特性.



图 1 继承:自然模型

Fig.1 Inherit: natural model

继承的这种层次化的描述既体现了复杂系统的自相似性,又体现了系统的层次性.层次观是系统科学的一个主要观点,任何复杂系统都是分层次的,层次之间互相联系又互相区别.不同层次之间的区别不只是量的区别,而且是质的区别.层次之间存在既有联系又有区别的辩证关系,不同层次之间的事物及其性质之间是有相互联系的,但是这种联系并不是简单的汇总或计算的关系.例如美国圣菲研究所提出的研究 CAS 系统的建模工具 SWARM 软件平台,它就是以多层次的建模方法为基础,可以嵌套,可以直接表示多层模拟,可以支持建立和描述多层

次的动态模型,从而实现对复杂系统的建模和仿真。SWARM 软件在多方面体现了面向对象的思想,层次化的描述更为明显。因此说,用继承机制描述系统时可减少复杂系统的复杂度。

### 3.4 多态性

面向对象另一个重要原则是多态性 (polymorphism)。多态性是指同一事件作用在不同对象身上有多种形式或实现方法。比如让对方说话,人可能说“你好”,狗会汪汪叫,猫会咪咪叫,但也可能就是不理你。

在面向对象系统中,多态性就是特定功能有多种实现方法。例如,我们可能要建立一个绘图系统,用户要画线、圆或者矩形时,系统发出绘画命令。系统中有许多形体,各有不同的绘图功能,因此,用户要画圆时,调用圆对象的绘图命令。利用多态,系统运行时确定要画的形体类型,而如果没有多态,则绘图功能的代码可能如下:

```
Function Shape .drawMe( )
{
    CASE Shape .Type
    Case“ Circle”
        Shape .drawCircle( ) ;
    Case“ Rectangle”
        Shape .drawRectangle( ) ;
    Case“ Line”
        Shape .drawLine( ) ;
    END CASE
}
```

利用多态,则只要对所画对象调用 drawMe( ) 函数,命令如下:

```
Function draw ( )
{
    Shape .drawMe( ) ;
}
```

每个形体(线、圆、矩形等等)用自己的 drawMe( ) 函数画图。

和面向对象的其他原则一样,多态的好处之一是易于维护。如果应用程序要画一个三角形,在非多态情形中,就要给 Shape 对象加一个新的 drawTriangle( ) 函数,Shape 对象的 drawMe( ) 函数也要修改成适应新形体的类型。而利用多态,则生成新的三角形对象,用 drawMe( ) 函数绘图,启动绘图操作的 draw( ) 函数根本不必改变。

霍兰在他的《隐秩序》一书中曾提到,所有的

CAS(Complex Adapt System)都具有多样性的特性,而多态性在描述 CAS 的多样性方面具有得天独厚的优势。

### 3.5 粒度

运用整体部分原则帮助观察者,使他们不至于遇到一些很大的东西就不知所措,这种原则叫做“粒度”。

在把建筑结构比例运用于一座特定建筑物时,整体部分这两项的关系必须加进第三项——观察者。这种三项关系叫做粒度。

通过粒度的概念,符号与策略中可以包含一些指导读者通晓一个大的 OOD 模型的方法。

## 4 基于 UML 的面向对象方法的复杂系统的模式构造 (Modeling construction with UML)

UML(统一建模语言, unified modeling language)是目前被广泛使用的面向对象系统的建模方法之一。它是一种可视化的建模方法,其优点是它可将模型中的信息用标准图形元素直观地显示出来, UML 可以开发几种不同的可视框图,表示系统的不同方面,如, Use Case 框图、Sequence 框图、Collaboration 框图、Class 框图、State Transition 框图、Component 框图、Deployment 框图,其中每个框图都有一定的用途和使用对象。我们这里主要用 Class 框图来描述经济复杂系统中的一个对象——银行模型。图 2 为银行模型的对象建立,图 3 为该模型的 Class 框图。

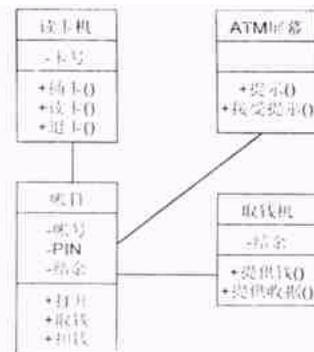


图 2 银行模型的对象建立

Fig.2 Creating an object of bank model

Class 框图显示了实现取钱使用案例中类之间的关系。这是用四个类完成的,读卡机、帐目、ATM 屏幕和取钱机。Class 框图中每个类用方框表示,分成三部分。第一个部分是类名,第二部分是类包含的属性,属性是与类相关联的信息,例如,帐目类包含三个属性:帐号、PIN 和结余。最后一部分包含类的操作,操作就是类提供的功能。帐目类包含四个操

作:打开、取钱、扣钱和验钱数.连接类的直线显示类之间的关系.例如,帐目类连接 ATM 屏幕类,因为两者要直接互相通信.读卡机与取钱机不连接,因为两者不能进行通信.

分析人员用 Class 框图显示系统细节.建筑师也用 Class 框图显示系统设计.如果一个类包含太多功能,则建筑师可以从这个 Class 框图中看出问题并将功能划分到多个类中.如果要相互通信的类之间没有建立关系,建筑师和开发人员也能从 Class 框图中看出.Class 框图可以显示每个使用案例中类的相互关系,也可以显示整个系统或子系统.

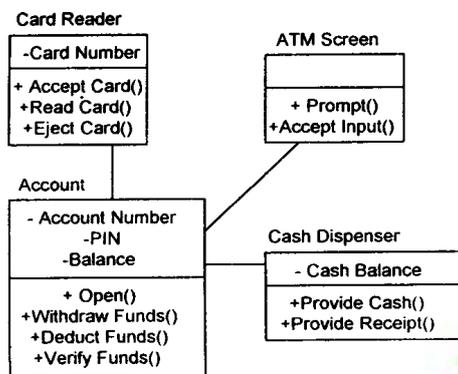


图3 银行模型的 Class 框图

Fig.3 The class diagram of bank model

由以上这个简单例子我们可以看出,利用 UML 来描述复杂系统,对其进行动态建模,能非常好地体现复杂系统的模式构造,也易于发现复杂系统的许多隐藏着的秩序,具有相当的可操作性.如果要对系统进行详细的建模,我们则可在各层次上对系统进行更细的描述.可以认为:当所开发的系统在所要求

的认识层次上有清晰的描述时,则系统的复杂性在这一层次上也就清楚了.这正是我们按照面向对象的机制层次地认识复杂系统的目的所在.

## 5 结束语 (Conclusion)

目前,对复杂系统的研究方兴未艾,各种复杂系统的研究方法也层出不穷地涌现出来.本文初步探讨了如何将面向对象机制引入到复杂系统的研究中来,同时选用了目前广泛使用的面向对象系统的建模工具 UML,对某一对象进行了模式构造.我们相信,随着更多领域研究人员的加入,及更多领域研究方法的引入,复杂系统的研究工作一定会获得越来越多的成果,同时它也会为众多的领域的研究发展提供新的思路、新的途径和新的方法.

## 参考文献 (References)

- 1 Coad P, Yourdon E. Object-oriented Design[ M]. Yourdon Press, 1990
- 2 Coad P, Yourdon E. Object-oriented Analysis[ M] Yourdon Press, 1990
- 3 Boggs W, Boggs B. Mastering UML with Rational Rose[ M]. SYBEX Inc. Press, 1999
- 4 陈禹. 层次——系统科学的一个重要范畴,系统科学与工程研究[ M]. 上海:上海科技教育出版社, 2001
- 5 霍兰著,周晓牧等译. 隐秩序——适应性造就复杂性[ M]. 上海:上海科技教育出版社, 2000

## 作者简介

尹春华(1966 - ),女,博士生.研究领域为复杂性科学与信息技术.

顾培亮,天津大学管理学院教授,博士生导师.研究领域为系统科学及复杂性科学.