第34卷 第4期
1995年 10月

中山大学学报(自然科学版)
ACTA SCIENTIARUM NATURALIUM
UNIVERSITATIS SUNYATSENI

Vol. 34 No. 4
Oct. 1995

试论对象和对象模型*

肖金声 阮文江 刘丹青
(中山大学计算中心，广州 510275)

摘 要 首先讨论面向对象技术的两个基本概念：对象和类，然后提出用于分析、设计和实现软件系统的对象模型.

关键词 对象，类，对象模型
分类号 TP311.52

自 Simula 67 问世以来，面向对象技术已有 20 多年的历史了. 对象及其有关概念被反复而广泛地使用着. 但在不同人的笔下或不同的场合，它们的含义却不尽相同，有必要作一反思，以分析差异，统一认识.

1 对 象

对象一词在自然语言中本就有个体（如，他的对象是个女秘书）、群体（如，这次的选送对象是……等 3 人）、种类（如，当前的侦破工作以……的人为对象），以及泛指（如，这家报社的采访对象是多方面的）等多种用法. 这种情况必然会反映到面向对象技术中来，应当予以区分.

在 OOPL 中，一般都正式或非正式地把对象定义为类变量或类实例. 这种先类而后对象的说法，可能是因为类是语法单位，而对象却不是的缘故. 值得注意的是，类变量和类实例有别：类变量可能随赋值的不同而代表不同的个体对象；类实例的含义则更接近个体对象.

今天，在系统分析和设计领域，没有人再采用先类后对象的说法，而是反过来. 我们来看几家有代表性的定义.

提倡对象建模技术（OMT）的 James Rumbangh 等人认为“对象是问题域中有意义的事物”[1]. 并解释说，它“对应用而言是一个概念、抽象或具有明确边界和意义的事物.”这明确表明，对象是一种泛称.

重视对象模型的 Booch 则说：“对象有状态、行为和标识；相似对象的结构和行为在它们的共同类中定义；术语实例和对象是可互换的”[2,3]. 这个定义对对象的本性进行了刻划，比泛泛的“事物”进了一步. 但它说得更多，涉及了类和对象的联系.

收稿日期：1994-10-28
* 国家 863 资助项目

Coad 和 Yourdon 的定义又有所不同："对象是问题域中事物的抽象……，是属性值及其相关服务的封装"[4]. 并认为实例是对象的同义词. 这里的属性值和相关服务相当于 Booch 所说的状态和行为. 这个定义说对象并非事物，而是事物的抽象. 并指出了抽象的内容，但用到了 OO 技术的一种处理手段—封装.

在回顾了对象的不同定义之后，笔者认为，定义对象概念应遵循以下 3 点：① 不涉及类与封装等后续概念；② OO 技术中的对象与自然语言中的对象有别；③ 对象一词宜只用于泛指.

故提议如下定义：

定义 1　对象是问题域中事物的数据抽象，它不仅有值，还具有特定的行为.

对象值的结构处决于问题域的需要 . 能在对象上执行的所有操作以及该对象要在别的对象上执行哪些操作就构成该对象的行为. "抽象"表明对象毕竟不是事物本身，而"数据抽象"则强调对象的数据性质.

我们的定义不提标识，因为标识有时是外界赋予的、临时的（是程序中指称实体对象的需要），不一定是对象所具有的特性.

此外，为在用语上区分泛称和个体，我们按 G. Booch 的说法把个体对象称为对象实体.

2　类

任一系统都有许许多多的对象实体，人们不可能逐个描述它们的结构和行为，于是就需要类来概括描述对象.

James Rumbangh 等人说："对象类描述具有相似性质的一组对象，这组对象具有共同行为、关系和语义"[1]. 类是对象类的简称.

Coad 和 Yourdon 则认为类是"对具有相同属性和服务集的对象，以及如何在该类上生成新对象的描述"[4]. 特地强调了新对象的生成.

Booch 的定义有点与众不同："类是共享共同结构和行为的对象集合[2,3]."

我们至少有两条理由不赞成类是对象集合的说法. 其一，对象须在程序运行时创建，而非在类中选取. 其二，需要将类用作对象的类型.

当今一些 OOPL 常把创建对象视为操作之一，没必要在类的定义中特别强调对象的生成. 故我们建议如下定义：

定义 2　类是对相似对象的描述，描述它们的共同结构和行为.

这就说明了类与相应对象的联系，又具体表明了描述的内容. 本文的定义没有涉及关系和语义，因为只有数据结构和行为才是类所刻意描述的.

在软件系统中，任何类的任一实例均是对象实体；反之，系统中没有一个对象实体不被某个类所描述.

类是 OO 技术中的关键概念，OO 技术对对象的处理都基于类.

3　对象模型

OO 技术提倡从问题域抽取对象，并按类进行描述. 此外，还对它们进行了如下几种处理：

3.1 封装 对象把数据和能在该数据上进行的操作紧密地组装在一起，这是OO技术与功能分解技术的本质区别. 但封装的含义还不止于此.

封装一般把类的描述分成两部分：接口与实现. 接口表达类的外观，供客户使用，实现则完成对客户隐藏的全部内情. 这种做法来自抽象数据类型.

3.2 类型化 OO技术还让类在OO语言中扮演过程语言中类型的角色，使得不同类型的对象原则上不能交换.

3.3 划分等级 把类和对象划分成等级的手段是继承和聚集. 继承使得子类能分享（一个或多个）父类定义的结构和行为，并允许子类补充新的内容或修订父类的某些内容. 继承的存在导致类型的多态.

与继承表现一般—特殊关系相比，聚集则是表现整体—部分关系. 它允许一个类的对象成为另一类的对象的一部分.

3.4 模块化 一个大系统中的类可能有成百上千个. 如果以类作为分解单位就太小了. 于是需要更大的单位—模块. 在模块中可以组装逻辑相关的类和较小的模块，以形成系统的物理层次. 模块的对外联系在于明确表明使用哪些输入类及提供哪些输出类. 模块应是内聚而松耦合的. 在许多程序语言中，模块常常是分别编译的单位.

类和对象加上上述的4种处理，就形成了OO技术的灵魂—对象模型[1]. OO技术按对象模型去分析、设计和实现软件系统.

参 考 文 献

1 Rumbangh J, Blaha M, Premerlani W, et al. Ojbect - oriented modelling and design, prentice hall. Englewood Cliffs, NJ (USA), 1991

2 Booch G. Object - oriented design with applications. The Benjamin/Cummings Publishing Company, Inc. 1991

3 Booch G. Object - oriented analysis and design with applications. The Benjamin/Cummings Publishing Company, Inc. 1994

4 Coad P, Yourdon E. Object - oriented analysis, 2nd ed. Yourdon Press, Englewood Cliffs, NJ, 1991

To Study Object and Object Model

*Xiao Jinsheng** *Ruan Wenjiang* *Liu Danqing*

Abstract This paper discusses firstly two basic concepts in object - oriented technic: object and class, and then provides a object model which is used to analyse, design and implement software system.

Keywords object, class, object model

[1] 这里所说的对象模型不是OMT中的静态对象模型，也不同于〔3〕中的对象模型.

* Computing Centre of Zhongshan University Guangzhou 510275