

# 对象形式语义模型\*

黄涛 冯玉琳

李京

(中国科学院软件研究所, 北京 100080) (中国科学技术大学, 合肥 230027)

**摘要** 在面向对象的软件构造中,对象被视为软件系统的基本构件. 本文给出一个对象形式语义模型,对象被定义为封装属性和行为的实体,对象行为必须满足给定的静态约束和时序约束;基于对象态射,本文给出对象聚合、特化、继承和对象类等概念在此语义模型框架中的解释.

**关键词** 对象语义模型,继承,聚合,特化.

在面向对象的软件构造中,对象被视为软件系统的基本构件. 在众多讨论中,对象被描述为抽象数据的推广,或封装数据和进程的载体<sup>[1]</sup>. 尽管人们已对面向对象技术作了广泛的研究,但至今仍没有一个普遍接受的面向对象概念的定义,其原因在于目前的面向对象方法缺乏严格的数学基础<sup>[2]</sup>.

这里,我们结合基于代数规范方法的代数模型<sup>[3]</sup>和基于状态约束的时序模型<sup>[4]</sup>提出对象的形式语义模型,给出诸如对象(object)、对象行为(behavior)、对象类(class)、对象继承(inheritance)、对象相互作用(interaction)、对象聚合(aggregation)、对象特化(specialization)等对象概念的形式语义解释.

作为数据封装的载体,对象体现了信息隐藏和封装的概念,对象的内部状态对外界是不可见的,外部世界只能通过对象的特性如对象属性值、动作序列等来感知对象的存在. 对象的特性可以由对象的属性来定义,但特性不等于属性. 对象属性依赖于建模的目的、观察点设置等. 如对一个产品,厂家、销售商和顾客用不同的观察,赋予产品不同的属性.

第1节将给出有关对象的一些基本概念及其形式化定义和解释. 第2节讨论对象之间关系,如 *is-a*, *is-part-of*, 并引入对象态射概念,用对象态射解释对象聚合、特化等. 第3节给出对象标识和对象类的定义,并区分继承和相互作用这两个对象构造概念.

## 1 对象实体

首先我们以队列为例说明有关对象的一些基本概念. 一个以集合  $E$  中元素为组成元素的队列类型可以看作是  $E$  中元素的有穷序列的集合  $S$ , 即  $S = E^*$ , 这个队列类型同时还附

\* 本文1993-08-21收到,1994-03-14定稿

作者黄涛,1965年生,1994年博士毕业于中国科学院软件研究所,主要研究领域为软件设计方法学,软件工程环境,语义理论,对象演算等. 冯玉琳,1942年生,研究员,博士生导师,中国科学院软件研究所所长,主要研究领域为系统模型和语义理论,软件设计方法学,软件工程环境等. 李京,1966年生,讲师,主要研究领域为软件设计方法学,软件工程环境.

本文通讯联系人:黄涛,北京 100080,中国科学院软件研究所

有一些操作,如  $front: S \rightarrow E$  (队列的头元素),  $rear: S \rightarrow E$  (队列尾元素),  $size: S \rightarrow Nat$  (队列长度),  $enter: S \times E \rightarrow S$  (在队列头部压入一个元素),  $leave: S \rightarrow S$  (从队列尾部移出一个元素). 根据面向对象的观点,一个队列并不是一个类型而是一个对象实例. 作为一个对象,队列具有自己的内部状态,队列的状态可以通过特定的窗口来观察,也可以被改变. 另外,队列展示其动态行为.

### 1.1 属性、观察和状态

设  $A$  是对象属性符号的集合,每个属性  $a \in A$ ,  $a$  的值域为  $codom(a)$ .

**定义 1.1.** 对象的一个观察 (Observation) 是这样的集合  $\{\langle a, d \rangle \mid a \in B, B \subseteq A, d \in codom(a)\}$ , 即一个观察给出了对象某些属性的值的集合.

**定义 1.2.** 对象的状态是在某一时刻该对象所有属性的观察.

由于对象属性并非在任何状态下均有意义,我们将在某个状态没有意义的属性视为无定义,也可观察. 以下我们将对象属性  $a$  与对  $a$  的观察不再加以区分,统一简称为属性  $a$ .

值得指出,属性是对象的一种特性. 对象属性指出一类可以赋给它的特定的值. 属性可以区分为状态主导属性和状态相从属性,状态主导属性决定对象的内部状态,而状态相从属性的引入则多是出于某种使用方便的考虑,如私有属性、导出属性等. 状态相从属性依赖于构成对象状态的主导属性,对象状态的改变亦可能导致这些相从属性值的改变.

### 1.2 约束

显然,并非所有的对象属性值的组合都可用来表示对象的状态,事实上只有部分对象属性值的组合是允许的.

**定义 1.3.** 对象约束是关于对象属性的限制或对象属性关系的描述.

对象约束也是一种对象特性. 有两类不同的对象约束:静态约束和动态约束. 静态约束表示对象属性之间的制约关系,规定了对象的合法状态,亦称状态约束,可以用一阶谓词逻辑公式表示,这种约束规则给出了对象所有可能状态集合到集合  $\{\text{“合法”}, \text{“不合法”}\}$  的一个映射. 动态约束表示对象合法状态之间的时序制约关系,亦称时序约束. 时序约束可以用时序逻辑公式来表示.

### 1.3 事件和行为

对象的属性和对象之间的关系构成了对象的静态结构. 然而对于对象而言更重要的是它的动态行为. 可以使用进程理论描述对象的动态行为,即对象是进程<sup>[5]</sup>,但进程理论不能描述进程之间的关系.

要描述对象的动态行为,动作 (action) 和观察是基本概念. 动作用于改变对象的内部状态而观察可以揭示对象的内部状态.

**定义 1.4.** 一个事件 (event) 是在某一瞬间发生的动作,导致对象状态变迁  $\langle s_1, s_2 \rangle$ ,  $s_1, s_2$  分别为动作前后对象所处的状态.

对象一旦初始建立,就在对象动作的作用下不断更新变化. 作用在初始对象上的一串事件序列,称为对象踪迹 (Trace). 对象踪迹记录了对象在一特定时间内的行为,决定对象演变的历史,即从对象初始状态出发的一串对象状态的序列. 由于观察不改变对象状态,对象状态由对象动作和动作作用前的状态完全决定. 因此,在考虑对象演变的历史时,可将对象事件序列和对象动作序列等同看待. 而对象在某一时刻的状态则完全由对象的初始状态和对

象踪迹所决定.

设对象的事件动作符号表为  $E$ .

**定义 1.5.** 对象行为是所有允许的对象踪迹的集合, 即,  $beh = (E, \Gamma)$  使得  $\Gamma \subseteq E^*$ ,  $\epsilon \in \Gamma$  且  $\Gamma$  满足该对象的特定(时序)约束.

对任意  $a, b \in E, A, B \subseteq E$ , 若  $A \cap B = \emptyset$  则:

$$((a \text{ perm}(A) \text{ perm}(B) b) \cup (a \text{ perm}(B) \text{ perm}(A) b)) \subseteq \Gamma \Rightarrow (a \text{ perm}(A \cup B) b) \subseteq \Gamma \quad (*)$$

这里  $\text{perm}(X)$  表示  $X$  上所有全排列的集合,  $a \text{ perm}(A) b$  表示  $A$  中事件可以在  $a$  与  $b$  之间以任意次序发生(或称并发), 上述(\*)表示如果  $A, B$  是这样两个可以以任意次序发生的“并发”事件集合, 那么  $A, B$  中所有事件也可以“并发”发生.

**定义 1.6.** 对象属性  $a \in A$ , 其属性值映射函数  $\alpha_a: E^* \rightarrow 2^{\text{codom}(a)}$  给出了任一合法事件序列下属性  $a$  可能取值的集合.  $\alpha = (\alpha_a)_{a \in A}: E^* \rightarrow 2^{\{(a,d) | a \in A, d \in \text{codom}(a)\}}$  称为对象映射函数. 特别地,  $\alpha_a(\epsilon)$  给出对象属性  $a$  的初始属性值,  $\alpha(\epsilon)$  给出初始状态下对象诸属性的值.

综上所述, 对象实体具有状态相关属性, 响应特定事件而导致状态变迁并展示其行为, 对象行为需满足给定约束(时序约束和状态约束).

**定义 1.7.** 对象实体是对象属性, 对象事件, 对象行为和对象映射函数所组成的四元组, 即  $ob = (E, A, \Gamma, \alpha)$ ,  $\Gamma \subseteq E^*$ . 特别地,  $obt = (E, A, \alpha)$  称之为对象模板.

## 2 对象态射

### 2.1 对象态射

对象之间存在各种各样的关系, 这些关系中最重要的一类是  $is-a$  和  $is-part-of$ , 即  $ob_1 is-a ob_2$  表示对象  $ob_1$  同时是另一对象  $ob_2$ ,  $ob_1 is-part-of ob_2$  表示  $ob_1$  是  $ob_2$  的一个组成成份. 所有这些对象关系可以通过对象态射这一概念阐述.

**定义 2.1.** 设对象实体  $ob_1, ob_2$ , 对象态射  $h: ob_1 \rightarrow ob_2$  是这样一对  $(h_E, h_A)$ , 其中:

$h_E: E_1 \rightarrow E_2$  满足行为态射条件(BMC):  $H_E^{-1}(\Gamma_2) \subseteq \Gamma_1$ , 这里

$$H_E^{-1}(\{\epsilon\}) = \{\epsilon\}$$

$$H_E^{-1}(\{e_1\}) = \text{perm}(h_E^{-1}(e_1))$$

$$H_E^{-1}(\{e_1 \dots e_n\}) = H_E^{-1}(\{e_1\}) \dots H_E^{-1}(\{e_n\}) \quad n \geq 1$$

$$H_E^{-1}(S) = \bigcup_{s \in S} H_E^{-1}(\{s\})$$

$h_A: A_1 \rightarrow A_2$  满足值域态射条件(CMC):  $\forall a \in A_2, \text{codom}_2(a) \subseteq \text{codom}_1(h_A^{-1}(a))$

和值态射条件(VMC):  $s_2 \in E_2^*, H_A^{-1}(a_2(s_2)) \subseteq a_1(H_E^{-1}(s_2))$

这里对于  $\delta \subseteq \{(a, d) | a \in A_2 \wedge d \in \text{codom}_2(a)\}$ ,  $H_A^{-1}(\delta) = \{(a, d) | a \in A_1 \wedge (h_A(a), d) \in \delta\}$

显然 BMC 意味  $ob_1$  中所有映射到  $ob_2$  的同一事件可以以任意次序发生即它们是“并发”的; 根据 CMC,  $ob_2$  属性的值域不大于  $ob_1$  相应属性的值域; VMC 意味着  $ob_1$  中属性映射到  $ob_2$  后, 将以同样的方式在  $ob_2$  中求值.

**定理 2.1.** 对象和对象态射构成一范畴, 称为对象范畴 OB.

证明: 略.

**定义 2.2.** 给定对象态射  $(h_E, h_A): ob_1 \rightarrow ob_2$ , 如  $h_E, h_A$  均为包含态射, 则称  $ob_2$  继承  $ob_1$  的

属性和行为.

### 2.2 对象聚合

对象可以复合成“更大的”对象以展示复合行为,称为对象聚合.聚合对象的行为依赖于它的组成成份对象.相互独立对象的聚合是很简单的,每个成份对象各自独立地决定新对象的行为特性,称为不交聚合;另一种聚合为并行聚合,新对象的行为特性取决于成份对象的动作交叉作用的结果<sup>[6]</sup>.

对象的相互作用可分为两类:共享存储和通讯,在我们的语义模型中前者对应于属性共享,后者对应于事件共享或调用.属性共享和事件共享是对象共享的特例.

定义 2.3. 对象实体  $ob_1, ob_2$  的不交聚合:  $ob_1 + ob_2 = (E_1 + E_2, A_1 + A_2, \Gamma_1 + \Gamma_2, \alpha_1 + \alpha_2)$ .

定义 2.4. 对象实体  $ob_1, ob_2$  的并行聚合:  $ob_1 \parallel ob_2 = (E_1 + E_2, A_1 + A_2, \Gamma_1 \parallel \Gamma_2, \alpha_1 + \alpha_2)$ .

这里,  $\Gamma_1 \parallel \Gamma_2 = \{\tau \in (E_1 + E_2)^* \mid \tau \uparrow E_1 \in \Gamma_1 \wedge \tau \uparrow E_2 \in \Gamma_2\}$ , 其中:  $\tau \uparrow X$  表示  $\tau$  受限于  $X$ , 不计  $\tau$  中所有不属于  $X$  的事件, 即:

$$\begin{aligned} \epsilon \uparrow X &= \epsilon \\ x\tau \uparrow X &= x(\tau \uparrow X), & \text{若 } x \in X \\ x\tau \uparrow X &= \tau \uparrow X, & \text{若 } x \notin X \end{aligned}$$

这里我们将传统的进程并行复合“ $\parallel$ ”扩充到对象聚合.更一般地,我们用 *is-part-of* 关系定义对象聚合.对象聚合为其组成对象参数的不相交复合,即事件符号表、属性符号表的不相交的并、行为的不相交并行复合以及属性值映射函数的相加.

定义 2.5.  $ob_1$  *is-part-of*  $ob_2$  当且仅当存在一个  $ob_1$  到  $ob_2$  的包含态射,  $ob_1$  称为  $ob_2$  的部分对象.

定理 2.2. 聚合对象的成分对象是聚合对象的部分对象.

分析 *is-part-of* 关系的态射条件可看出: BMC 意味着聚合对象的每个踪迹可受限为其成份对象的一合法踪迹,这是交叉作用的基本要求; CMC 指出聚合对象的属性的值域是其某个成份对象相应属性的值域的子集,也即是说当我们由部分构造整体时,可进一步限制属性值域;而 VMC 指出聚合对象所引入的某个成份的属性只受相应对象的事件的影响.

### 2.3 对象特化

现在我们讨论  $ob_2$  *is-a*  $ob_1$  的涵义,即  $ob_2$  是  $ob_1$  的特化.直观上讲,我们所希望的是,如果我们抽象掉一些细节,  $ob_2$  可以被视作  $ob_1$ .例如,一个学生同时是一个人,前者可以有更多的属性和事件,更多的特殊行为和属性求值,但所有这些应该可以被抽象掉而被视作一“简单”的人.

定义 2.6.  $ob_2$  *is-a*  $ob_1$  当且仅当存在一个  $ob_1$  到  $ob_2$  的包含态射,  $ob_2$  称为  $ob_1$  的特化.

值得指出的是,这是一个很强的定义.根据 VMC,新引入的动作不可以修改原来对象定义的属性.然而,继承的动作可能具有更多的作用(但对继承的属性作用不变),这一点也是有用的,它为方法重载(method overriding)提供了基础.

## 3 对象类

### 3.1 对象标识

对象可以通过其外部行为为外界所观察,现实世界中许多对象展示相同的外部行为,但它们分别作为不同的对象实体而存在.也就是说对象具有这样的一种特性使它得以唯一地区分于其它对象,我们称之为对象标识(identity).对象标识使得我们可以清晰地区分和刻划诸如继承、相互作用等对象概念;相互作用刻划了不同对象之间的关系,而继承则刻划了同一对象的不同侧面(aspect)之间的关系.

**定义 3.1.** 对象标识是一(原子)项,它使得该对象唯一地区分于其它对象.

对象标识提供了区分对象的机制,这是对象的一项重要特性.

令  $I$  为所有对象标识的集合(称为对象标识空间),  $T$  为对象实体的集合.

**定义 3.2.** 对象侧面是这样二元组  $(b, u)$ , 这里  $b \in I, u \in T$ .

对象侧面  $(b, u)$  表示标识为  $b$  的对象具有实体  $u$  的行为. 一个对象可以有多个侧面,如一个人可以是一学生也可以是一病人,甚至是它们的组合.

**定义 3.3.** 令  $(b, u), (c, v)$  是两个对象侧面,  $h: u \rightarrow v$  是一对象态射, 则  $h: (b, u) \rightarrow (c, v)$  称为侧面态射.

**定义 3.4.** 一侧面态射  $h: (b, u) \rightarrow (c, v)$ , 若  $b=c$  且  $h$  是包含态射, 则称为继承态射; 若  $b \neq c$ , 则称为相互作用态射.

如果不考虑对象标识, 根据定义 2.5、2.6,  $ob_1$  is-a  $ob_2$  当且仅当  $ob_2$  is-a part-of  $ob_1$ . 但标识是对象的一项特性, 同标识一起考虑, 关系 is-a 和 is-a part-of 就不可等同而论了.

**定理 3.1.** 对象实体在继承态射意义下构成一格  $\Delta$ .

证明: 继承态射是一包含态射, 它满足自反性、传递性和反对称性, 因而是一序关系. 令  $O_i$  为仅含所有对象共有特性的对象实体,  $O_b$  为包含所有特性的对象实体, 对任意对象实体  $ob$ , 存在包含(继承)态射  $h: O_i \rightarrow ob$  和  $h': ob \rightarrow O_b$ , 则  $\Delta$  为一格,  $O_i, O_b$  分别为其上下顶.

**定义 3.5.** 设有对象侧面  $(b, u)$  和  $(b, v)$ ,  $(b, v)$  称为  $(b, u)$  的导出侧面当且仅当存在一继承态射  $v \rightarrow u$ , 或存在一  $(b, u)$  的导出侧面  $(b, u')$  且有一继承态射  $v \rightarrow u'$ .

从实现讲, 要创建一对象必须提供相应的(由系统生成的唯一的)对象标识  $b$  和对象实体  $u$ , 我们称相应的对象侧面  $(b, u)$  为该对象的初始侧面. 一对象包含它的初始侧面及其所有导出侧面.

**定义 3.6.** 对象是对象初始侧面及其所有导出侧面的总称.

因为初始侧面唯一地标识了对象和对象行为, 我们可以定义对象为其初始侧面.

### 3.2 类

一般而言, 对象并不单独存在, 它们通常是作为某个“类”的成员而存在.

**定义 3.7.** 对象类  $C = (a, (E^C, A^C, \Gamma^C, \alpha^C))$  是一对象(初始侧面), 满足:

- $T, Id \in A^C, \text{codom}(T) \subseteq \text{Set}(OB), \text{codom}(Id) \subseteq 2^I$ , 其中  $\text{Set}(OB)$  为对象实体集;
- 对任意  $s \in \Gamma^C, \alpha_{\Gamma}^C(s) = \alpha_{\Gamma}^C(\epsilon)$ ;
- 一对象标识  $i \in ID, ID \in \alpha_{Id}^C(s), s \in \Gamma^C$  当且仅当存在有对象侧面  $(i, u), u \in \alpha_{\Gamma}^C(s)$ .

类对象提供诸如插入、删除成员等动作以及类中当前所包含的对象个数、对象标识集等观察. 上述定义的最后条件实际上是类对象的静态约束.

**定义 3.8.** 一对象称为对象类  $C = (a, (E^C, A^C, \Gamma^C, \alpha^C))$  的一个实例当且仅当其对象标识  $i \in ID, ID \in \alpha_{Id}^C(s), s \in \Gamma^C$  且存在一对象侧面  $(i, u), u \in \alpha_{\Gamma}^C(s)$ .

**定义 3.9.** 对象类  $C_2$  称为  $C_1$  的子类当且仅当  $C_2$  的每个实例是  $C_1$  的实例.

显然,若对象类  $C_2$  是  $C_1$  的子类,则  $\alpha_{C_2}^{\epsilon}(\epsilon) \text{ is-a } \alpha_{C_1}^{\epsilon}(\epsilon)$ . 通常,我们称类  $C_2$  继承  $C_1$  是指类  $C_2$  的每个实例的对象侧面继承相应  $C_1$  实例对象侧面的属性、行为等.

#### 4 结束语

本文旨在讨论面向对象的形式语义模型,给出对象、对象行为、对象约束、继承、相互作用、对象类等概念的语义解释,为我们清晰地理解面向对象技术提供理论基础. 由于篇幅有限,许多对象概念未能进一步讨论,如对象精化(reification)提供在已有对象的基础上构造新对象的机制;对象抽象(abstraction),作为对象继承的逆构造,提供隐藏细节的机制等. 仅提供对象的语义模型是不够的,进一步研究工作包括建立基于语义模型的规范描述、实现和验证,以求形成完整的支持面向对象系统的形式化开发. 本文所提出的语义模型为规范描述语言、设计语言以及正确性验证的演算提供了良好的语义框架.

#### 参考文献

- 1 Nierstrasz O M. A survey of object oriented concepts. In: Kim W, Lochovsky F ed. Object—Oriented Concepts, Databases, and Applications, Addison—Wesley, Reading, MA, 1988.
- 2 Ehrlich H D, Sernadas A, Sernadas C. From data types to object types. J. of Information Processing and Cybernetics EIK, 1990, 26(1/2):33—48.
- 3 Ehrig H, Mahr B. Fundamentals of algebraic specifications 1: equations and initial semantics. Springer—Verlag, 1985.
- 4 Emerson E. Temporal and modal logic. In: Leeuwen J ed. Formal Models and Semantics, Elsevier Science Pub., 1990. 995—1072.
- 5 Jungclaus R, Saake G, Sernadas C. Formal specification of object systems. In: Abramsky S, Maibaum T eds. Proc. TAPSOFT'91, LNCS 494, Springer, 1991. 60—82.
- 6 冯玉琳,李京,黄涛. 对象语义理论和行为约束推理. 计算机学报, 1993, 16(11):823—828.

## A FORMAL SEMANTIC MODEL FOR OBJECT

Huang Tao Feng Yulin

(Institute of Software, The Chinese Academy Sciences, Beijing 100080)

Li Jing

(University of Science and Technology of China, Hefei 230027)

**Abstract** Objects are considered as the basic system blocks in object oriented software construction. This paper contributes to establish a formal semantic model for objects. An object is defined as an entity endowed with attributes and behavior satisfying some static and temporal constraints. Based upon object morphism, the notion of object aggregation, specialization, inheritance and object class etc. are interpreted in the model further.

**Key words** Object semantic model, inheritance, aggregation, specialization.