对象系统的组合交互计算*

钱 军, 黄 涛, 冯玉琳

(中国科学院 软件研究所 计算机科学开放研究实验室,北京 100080);
(中国科学院 软件研究所 对象技术中心,北京 100080)
E-mail: qianjun@softel.com.cn
http://www.iscas.ac.cn

摘要: 对象系统的计算是一个开放的、动态并发交互的过程,由系统与外部环境的动态交互以及系统内部各组件间的协同工作来完成.对象系统的行为变化是系统内、外因有机统一的具体表征.因此不能把对象系统的静态组合构造和动态计算行为当作两个孤立的个体分而治之.从这个统一原则出发,研究了基于组合构造模型的对象系统的交互计算.组合构造模型设计将对象系统的主要计算特征纳入一个统一的抽象基调之中,通过定义基调上的一组关系,把对象系统的层次结构和内部协同工作有机地结合在一起.在此基础上,给出了一个区分动作类型的交互演算(S/R 演算),描述了组合对象系统的动态行为和交互,较好地刻画出对象封装性、对象计算的事件驱动性、动态绑定以及对象系统内部的交互与并发等固有特征.

关键词: 对象系统;组合构造模型;S/R 演算;并发;交互
中图法分类号: TP311 **文献标识码**: A

传统的结构化程序设计强调自顶向下逐步求精的过程.若仅从系统功能的精化分解去考虑,将造成系统内部结构庞杂而且独立性差,系统不易维护和扩展,灵活及复用性差.面向对象的组合程序设计强调系统设计模块化,采用自顶向下求精和自底向上组合相结合的双向设计过程,较好地解决了系统的独立性、扩展性和复用性等问题.如果说算法可计算系统是对传统计算机软件系统的抽象,那么分布式交互计算系统则是对具有并发、交互行为的分布式计算系统的一种抽象.随着计算机应用范围的不断扩大,特别是 Internet 的出现和发展,网络计算逐步成为计算机应用系统的主流.如此大规模的计算机软件系统的计算过程,是由系统与外部环境的动态交互以及系统内部各组件间的协同工作来完成的.

基于网络分布环境的对象系统是目前最主要的分布式计算系统之一.基于对象技术的组合软件开发方法是当今软件工程领域应用的主流技术.对象系统的计算是一个开放的、动态并发交互的过程,由系统与外部环境的动态交互以及系统内部各组件间的协同工作来完成.对象系统的行为变化是系统内因和外因有机统一的具体表征,因此不能把对象系统的静态组合构造和动态计算行为当作两个孤立的个体分别研究对待,这是本文工作的一个基本出发点.另一方面,对象技术的发展与对象理论的研究很不平衡.理论研究尚不成熟,因此,其发展远远滞后于工业界软件技术的突飞猛进.这种发展的不平衡将会直接影响软件系统的质量,因此,对对象系统的抽象规范与实现一致性验证具有重要的理论价值和实际意义.

* 收稿日期: 1999-10-28; 修改日期: 2000-05-19
基金项目: 国家重点基础研究发展规划 973 资助项目(G1998030404);国家自然科学基金资助项目(69833030)
作者简介: 钱军(1965-),男,山西大同人,博士,助理研究员,主要研究领域为分布计算,对象技术,计算机形式语义;黄涛(1965-),男,江苏淮阴人,博士,研究员,博士生导师,主要研究领域为软件工程,分布计算,对象技术,形式化方法;冯玉琳(1942-),男,江苏启东人,博士,研究员,博士生导师,主要研究领域为软件工程,分布计算,对象技术,形式化方法.

通常采用进程代数或状态机等工具来研究对象系统的计算过程,通过操作语义和互模拟来刻画对象系统的计算过程和系统之间的行为等价.其代表性的工作有:

(1) 并发面向对象程序的转换.如 D. Walker 等人的并发面向对象程序的转换,给出一种并发面向对象语言的 CCS 及 π 演算语义[1],C. L. Talcott 的 Actor 语言的复合语义[2],等等.

(2) 对象演算.如 O. Nierstrasz 的 Object Calculus[3],K. Honda 和 M. Tokoro 等人的异步通信演算[4],K. Fisher 等人用 λ 演算讨论对象计算,结果也很好[5].

(3) 状态机.如 N. A. Lynch 和 M. R. Tuttle 的 Input/Output 自动机[6],G. Agha 的 Actor 模型[7],以及利用 Petri net 来讨论对象系统的真并发行为[8].

(4) ISO 开发的分布式系统的描述工具 LOTOS.

通常采用时序逻辑与代数规范相结合的方式研究对象系统的行为规范,通过指称语义刻画对象系统的生命期规范.这方面的研究对系统的形式化描述和验证起到了不小的作用.其中较为典型的有:

(1) M. Abadi 和 L. Cardelli 对对象类型和继承的讨论;

(2) H-D. Ehrich 等人的分布时序逻辑和对象规范[9];

(3) Object-Z 和 OOVDM;

(4) 冯玉琳、黄涛等人的对象语义模型和对象演算[10~12].

本文试图将对象系统的计算过程,特别是对象交互计算特征的研究,纳入一个统一的抽象概念模型,即首先建立对象系统的组合构造模型,再在这个统一的概念框架下,通过一个区分动作类型的类 CCS 演算(S/R 演算)来描述组合对象交互计算语义,刻画对象封装性、对象计算的事件驱动性、动态绑定以及对象系统内部的交互与并发特征等.

本文第 1 节给出对象系统的组合构造模型.第 2 节构造一个区分动作类型的 S/R 演算,并给出演算的操作语义.第 3 节基于组合对象模型和 S/R 演算,描述对象系统行为的动态交互计算过程.第 4 节进行总结.

1 对象系统的组合构造模型

为了能够准确地描述对象系统的动态行为变化过程,必须对这样的系统进行形式抽象.对象系统由对象自底向上组合构造而成,上层对象称为下层对象的组合对象(composite object),下层对象称为上层对象的组成对象(component object),并且上层对象是对下层对象的聚合(aggregation)和封装(encapsulation).作为对象系统的基本单元,对象具有唯一标识(identifier),对象状态的改变只能通过对象方法(method)来实施.组合对象的方法可以抽象为组成对象的方法序列,称为方法传播(propagating),即通过组成对象方法之间的传播来完成组合对象方法的实施.基于以上分析,我们给出组合构造模型的如下定义:

定义 1. 称五元组 $\Sigma^C=(I,A,IA,AR,\partial)$ 为组合对象基调(signature),如果

- $I=\{i,j,\dots\}$ 为一组对象标识集合;
- $A=\bigcup_{i\in I}A_i=\{a,b,\dots\}$ 为一组动作集合的并;
- $IA\subseteq I\times A$ 为方法集合,满足:

(IA1) $\forall i\in I,\forall a\in A,(i,a)\in IA$ 当且仅当 $a\in A_i$;

- $AR\subseteq I\times I$ 为 I 上的一个抽象聚合关系,满足:

(AR1) $\forall i\in I,(i,i)\in AR$

(AR2) $\forall i,j\in I$,若$(i,j)\in AR$,且$(j,i)\in AR$,则$i=j$

(AR3) $\forall i,j,k\in I$,若$(i,j)\in AR,(j,k)\in AR$,则$(i,k)\in AR$

(AR4) $\forall i,j,k\in I$,若$(i,k)\in AR,(j,k)\in AR$,则$(i,j)\in AR$,或$(j,i)\in AR$

(AR5) $\forall i,j\in I,\exists k\in I$,使得$(k,i)\in AR$,且$(k,j)\in AR$

• $\partial:IA\to IA^*$为IA到IA^*的一个传播映射(propagating map),满足:

(PM1) $\forall (i,a)\in IA,\partial((i,a))=(i_1,a_1);(i_2,a_2);\ldots;(i_n,a_n)$,

其中, $i_k\in\downarrow i,(i_k,a_k)\in IA,1\leqslant k\leqslant n$.

这里,$n=\text{length}(\partial((i,a)))$表示传播方法序列的长度.$\downarrow i=\sup\{j\mid j\neq i,(i,j)\in AR\}$表示组合对象标识$i$的组成对象标识的集合.

下面,我们对定义1作些说明.虽然基调Σ^C的对象标识集和动作集是无结构的,但是Σ^C刻画了一个对象系统的组织结构,这由Σ^C中的二元关系IA和AR来确定.关系IA表明属于某个特定对象的特定动作集,而关系AR用一组条件(AR1)~(AR5)表明组合对象的抽象聚合.从某种意义上讲,上层组合对象是对其下层组成对象集的抽象.本文在基调Σ^C中引进对象方法传播映射,使对象系统的结构层次和内部协同工作有机地结合在一起,这是我们给出的组合构造模型的新颖之处.对象系统的组合构造模型将作为本文研究系统动态行为特性的一个统一的抽象概念框架,贯穿论文始终.

性质1. 给定组合对象基调$\Sigma^C=(I,A,IA,AR,\partial)$,则抽象聚合关系AR为一个偏序关系.

约定. 若S表示集合,$\leqslant$表示S上的一个偏序关系,则用$\sup\{S,\leqslant\}$表示集合S在偏序$\leqslant$下的极大元素集合.用$\text{sub}\{S,\leqslant\}$表示集合$S$在偏序$\leqslant$下的极小元素集合.

定义2. 给定组合对象基调$\Sigma^C=(I,A,IA,AR,\partial)$,把$I$在关系AR下的集合$\sup\{i\in I\}$和$\text{sub}\{i\in I\}$分别称为根对象标识集合和叶对象标识集合.

定义3. 给定组合对象基调$\Sigma^C,\forall i\in I$,

$i\in I-\sup\{i\in I\}$,称$\uparrow i=\text{sub}\{j\mid j\neq i,(j,i)\in AR\}$为$i$的(上层)组合对象标识集合,

$i\in I-\text{sub}\{i\in I\}$,称$\downarrow i=\sup\{j\mid j\neq i,(i,j)\in AR\}$为$i$的(下层)组成对象标识集合.

性质2. 给定组合对象基调Σ^C,则

• $\forall i\in I-\sup\{i\in I\}$,$\uparrow i$具有唯一元素,

• $\forall i,j\in I-\text{sub}\{i\in I\}$,且$i\neq j$,则$\downarrow i\cap\downarrow j=\varnothing$.

定义3和性质2严格给出了(上层)组合对象和(下层)组成对象的定义和基本特征,根对象作为一个组合对象基调的"最大"(最高层)对象,封装了其所有组成对象;叶对象作为一个组合对象基调的"最小"(最底层)对象,被其上层对象封装.

为简便起见,在不引起混淆的情况下,我们把$\partial((i,a))$记为$\partial(i,a)$,表示(i,a)的传播序列,把$(\partial(i,a))_k$记为$\partial_k(i,a)$,表示传播序列$\partial(i,a)$上的第k个元素.

我们作以下约定:

• $\sigma=\sigma_1;\sigma_2;\ldots;\sigma_n$为一个字符串,称$\sigma_i$为$\sigma$的第$i$个字符,$^i\sigma$为$\sigma$去掉前$i-1$个字符$\sigma_1,\sigma_2,\ldots\sigma_{i-1}$以后的字符串,$head(\sigma)=\sigma_1$为$\sigma$的首字符,$tail(\sigma)=^1\sigma$为$\sigma$的尾字符串.显然,

$$head(^i\sigma)=\sigma_i.$$

• $\rho_I:IA\to I,\rho_A:IA\to A$分别为IA在$I$和$A$上的投影映射,即$\forall (i,a)\in IA,\rho_I((i,a))=i$,$\rho_A((i,a))=a$.

性质3. 给定组合对象基调$\Sigma^C=(I,A,IA,AR,\partial),\forall i\in I$,则

(PM2)
$$\partial(i,a)=head(\partial(i,a));tail(\partial(i,a));$$
$$\partial_k(i,a)=(\partial(i,a))_k;$$
$$\rho_I((i,a))=i,\rho_A((i,a))=a\in A_i;$$
$$\rho_I(\partial_k(i,a))\in\downarrow i,(\rho_I(\partial_k(i,a)),\rho_A(\partial_k(i,a)))\in IA.$$

定义4(对象系统基调). 设对象系统S由n个组合对象构成,并且这n个组合对象基调分别为$\Sigma_k^c=(I^k,A^k,IA^k,AR^k,\partial^k),1\leqslant k\leqslant n$,则称

$$\Sigma^S=\Sigma_1^c\cup\Sigma_2^c\cup\ldots\cup\Sigma_n^c$$

为对象系统基调. 其中,Σ^S可以表示为一个五元组(I,A,IA,AR,∂),满足:

- $I=\bigcup_{k\in\{1,n\}}I^k$为对象标识集合;
- $A=\bigcup_{k\in\{1,n\}}I^k$为动作集合;
- $IA=\bigcup_{k\in\{1,n\}}IA^k\subseteq I\times A$为满足(IA1)的方法集合;
- $AR=\bigcup_{k\in\{1,n\}}AR^k\subseteq I\times I$为满足(AR1),(AR2),(AR3),(AR4)的抽象聚合关系;
- $\partial:IA\to IA^*$为IA到IA^*上的满足(PM2)的传播映射,且

$$\forall\ (i,a)\in IA^k,\partial(i,a)=\partial^k(i,a);$$

- $n=|\sup\{i\in I\}|$,表示集合$\sup\{i\in I\}$的模(元素个数).

注意到,在基调Σ^S下,关系AR不满足(AR5),因此$\sup\{i\in I\}$集合中的元素可能不再唯一.

2 S/R演算

我们构造一个区分交互动作类型及传值类型的类CCS演算[13],即S/R演算,用于刻画组合对象的动态行为变化过程. 对象计算是一个动态的、并发交互的过程,不能脱离一个对象所处的环境而孤立地看待其内部行为变化. S/R演算既要能够描述交互,又不能使对象失去组合的独立性和封装性. S/R演算使用尽可能少的交互动作(输入与输出),同时加强对传值的类型及条件检查,使对象能够在与环境的并发交互下保持自己的特性.

对象之间通过发送消息进行交互,存在多种交互形式. 有的交互在对象系统的组合构造过程中就已静态绑定,而有的交互则是在对象系统的计算过程中动态绑定的. 对象之间的交互大致可分为同步消息传递和异步消息传递两大类,交互可以一对一进行,也可以一对多、多对一以及多对多地进行.

令$Sorts=\{s,\ldots\}$为类型集合(包括简单类型和构造类型),$Id=\{i,j,\ldots\}$表示对象标识集合,$Act=\{a,b,\ldots\}$表示对象动作集合.

定义5. $M=\{m,\ldots\}$表示消息集合,其中$m=(m_1,m_2,m_3,m_4)$

$m_1\in Id^*$为对象标识或对象标识串,表示消息源;

$m_2\in 2^{Id}$为对象标识集合,表示消息目标;

$m_3\in Act^*$为对象动作或对象动作串,表示自顶向下的动作传播;

$m_4\in Id\times s$(构造类型).

这里,我们对消息四元组$m=(m_1,m_2,m_3,m_4)\in Id^*\times 2^{Id}\times Act^*\times(Id\times s)$的定义作些说明. 消息源$m_1\in Id^*$表示为一个对象标识或对象标识串,既说明消息的源路径,又刻画了组合对象对组成对象的封装. 消息目标$m_2\in 2^{Id}$表示为一个对象标识集合,支持一对多发送消息. $m_3\in Act^*$为对象动作或对象动作串,表示由交互触发的、自顶向下的动作传播. m_4属于构造类型$Id\times s$的参数.

可以通过消息源 m_1(对象标识串)中各对象标识之间的相互次序来确定消息的类型(是请求,还是结果返回等),也可以通过 m_4 表明消息类型.

定义 6. 令 $Interaction=Input\cup Output\cup Internal$ 为交互动作名集合,其中 $Input=\{R\}$ 为输入动作集合,$Output=\{S\}$ 为输出动作集合,$Internal=\{r,w,\ldots\}\cup\{\tau\}$ 为内部交互动作集合.

定义 6 区分交互动作类型是为了强调每个对象的行为均受其所处环境的制约.输入动作虽然局部不可控制,但是服从外部环境的管理.而输出动作和内部动作则是由内部控制的.

定义 7. 令 $A,\ldots$ 表示常量交互计算进程(interaction computation process),$X,\ldots$ 表示变量交互计算进程,$E=\{P,Q\ldots\}$ 表示所有交互计算进程表达式(含变量)的集合,$R\in Input$,$S\in Output$,$c\in Internal$,$m\in M$,x 表示 M 上的变量,则交互计算进程表达式递归定义如下:

(1) 0——终止 ICP 进程;

(2) $R(x:M).P$——接收消息的 ICP 进程;

(3) $[\psi:\text{boolean}]S(m:M).P$——发送消息的 ICP 进程;

(4) $[\psi:\text{boolean}]c.P$——内部计算 ICP 进程;

(5) $P+Q$——选择;

(6) $P|Q$——复合;

(7) $A=_{\text{Def}}P$——递归.

ICP 为交互计算进程的缩写.0 表示什么也不做的 ICP 进程.$R(x:M).P$ 表示在执行完输入动作 $R(x:M)$ 后执行 P 进程,输入动作的特征是无条件地接收消息.$[\psi:\text{boolean}]S(m:M).P$ 表示输出动作受局部的控制,控制由条件 $\psi:\text{boolean}$ 刻画,在条件满足时可以执行输出动作 $S(m:M)$,然后执行 P 进程."+"为选择算子,交互计算进程 $P+Q$ 表示要么执行 P,要么执行 Q."|"为选择算子,交互计算进程 $P|Q$ 表示 P 和 Q 要么交错执行,要么并发执行.$A=_{\text{Def}}P$ 定义交互计算进程递归表达式.

定义 7 给出的是一般的交互计算进程定义,通过这样的交互计算进程并不能立即看出它是属于哪个对象的.如果注意到每个对象均有唯一标识的特性,就不难发现解决这一问题的途径.定义 8 正是利用对象标识对交互计算进程进行了语义扩充.

定义 8. 令 $Id=\{i,j,\ldots\}$ 表示对象标识集合,$E=\{P,Q\ldots\}$ 为交互计算进程 ICP 的集合,则对象交互计算进程 obICP 的定义为

$$obICP:=ICP@i.$$

其中 $ICP@i$ 表示对象标识为 i 的某个对象的交互计算进程.

下面给出 S/R 演算的操作语义.

定义 9. 令 $P,Q,\ldots$ 为 ICP,$R\in Input$,$S\in Output$,$c\in Internal$,$\alpha,a,b\in Interaction$,$\psi,\varphi,\psi_1,\psi_2\in Boolean$,则 S/R 演算的操作语义"$\rightarrow\langle condition,interaction\rangle\rightarrow$"是由以下转移规则生成的:

Asynchronous-message-sent rules (Send)

$$[\psi]S(m).P@i\rightarrow\langle\psi,S(m)\rangle\rightarrow P@i|S(m)@m2$$

$$[\psi]S(m1,m2).P@i\rightarrow\langle\psi,S(m1)\rangle\rightarrow P@i|S(m1)@m1_2|S(m2)@i$$

$$[\psi]S(m1,m2).P@i\rightarrow\langle\psi,S(m2)\rangle\rightarrow P@i|S(m1)@i|S(m2)@m2_2$$

$$[\psi]S(m1,m2).P@i\rightarrow\langle\psi,S(m1)|S(m2)\rangle\rightarrow P@i|S(m1)@m1_2|S(m2)@m2_2$$

Message-Receive rules (Receive)

$R(x).P@i \rightarrow (\text{true}, R(x\sigma)) \rightarrow P\sigma@i$

若 $i \in L \subseteq Id$,则 $S(m)@L \mid R(x).Q@i \rightarrow \langle \tau \rangle \rightarrow Q[m/x]@i$

Computation-transition rules (Comp)

$[\psi]a.P@i \rightarrow \langle \psi, a \rangle \rightarrow P@i$

若 $P@i \rightarrow \langle \psi, a \rangle \rightarrow P'@i$,则 $P@i + Q \rightarrow \langle \psi, a \rangle \rightarrow P'@i$

若 $P@i \rightarrow \langle \psi, a \rangle \rightarrow P'@i$,则 $Q + P@i \rightarrow \langle \psi, a \rangle \rightarrow P'@i$

若 $P@i \rightarrow \langle \psi, a \rangle \rightarrow P'@i$,则 $P@i \mid Q \rightarrow \langle \psi, a \rangle \rightarrow P'@i \mid Q$

若 $P@i \rightarrow \langle \psi, a \rangle \rightarrow P'@i$,则 $Q \mid P@i \rightarrow \langle \psi, a \rangle \rightarrow Q \mid P'@i$

若 $P@i \rightarrow \langle \psi, a \rangle \rightarrow P'@i, Q@i \rightarrow \langle \zeta, b \rangle \rightarrow Q'@i$,则 $(P \mid Q)@i \rightarrow \langle \psi \wedge \zeta, a \mid b \rangle \rightarrow (P' \mid Q')@i$

若 $P@i \rightarrow \langle \psi, a \rangle \rightarrow Q@i, A =_{\text{Def}} P$,则 $A@i \rightarrow \langle \psi, a \rangle \rightarrow Q@i$

说明:

(1) 转移条件 ψ 说明对象交互计算进程的输出动作受怎样的局部(条件)的控制. 只有在条件 ψ 为真的时候,$[\psi]S(m).P@i$ 才能按 $S(m)$ 动作发生转移;当 ψ 为假时,$[\psi]S(m).P@i$ 为一个什么都不做的进程.

(2) 在条件 ψ 为真的条件下,$S(m).P@i$ 执行一个异步消息发送动作 $S(m)$,并演变为两个并发进程 $P@i \mid S(m)@m_2.P@i$ 对应于 $S(m).P@i$ 的后继计算,$S(m)@m_2$ 对应于一个异步消息发送和传递过程. $S(m)@m_2$ 为定义 8 的扩充,表示异步消息 m 已达可能的目的集合,其中 m_2 为消息 m 的目标分量.

(3) 由 Receive 规则可知,对象交互计算进程的输入动作不受局部(条件)的控制.

3 对象系统的组合交互计算

对象系统组合构造模型通过对象系统基调中的二元关系,强调了任何一个对象的行为都是受其所处环境制约的;通过引进对象方法传播映射,使对象系统的结构层次和内部协同工作有机地结合在一起. 在这个统一的抽象概念框架下,通过采用 S/R 演算,可以描述基于对象组合构造模型的对象系统的动态行为变化过程.

Object computing=environment+control+interaction①

①对象计算=环境+控制+交互.

Fig. 1 Component object model

图1 组合对象模型

对象计算的要素有哪些?对象计算的过程如何表示?图 1 用图示给出了一个对象计算模型. 环境、控制和交互是对象计算的 3 个要素. 图中标识 i 表示组合对象;标识 $j_1, j_2, \ldots, j_N$ 表示组合对象 i 的组成对象,即 $\forall j_k$, $1 \leqslant k \leqslant N$, $j_k \in \downarrow i$;$Env[i]$ 是对象 i 的环境;$MC[i]$ 是对象 i 的计算过程控制器,负责与对象外部环境发生交互,并通过方法传播序列的定义,控制对象系统内部的计算过程.

下面,我们基于对象组合构造模型所定义的统一框架,通过 S/R 演算来描述组合对象交互计算的诸要素.

3.1 环 境

给定对象系统基调 $\Sigma^s = (I, A, IA, AR, \partial)$,$\forall i \in I$,我们把 i 的环境描述为集合 $Env[i]$,如果

- i 为非根对象标识,则 $Env[i] = \uparrow i$;
- i 为根对象标识,则 $Env[i] = \sup\{i \in I\} - \{i\}$.

3.2 过程控制器

给定对象系统基调 $\Sigma^S=(I,A,IA,AR,\partial)$,$\forall\ i\in I$,我们把 i 的计算过程控制器描述为对象交互计算进程(*obICP*)$MC[i]$,如果

· 若 i 为非叶对象标识,则 $MC[i]=Console@i\mid\prod_{j\in\downarrow_i}MC\lceil j\rceil$.

这里,Console 为一个 ICP,负责对象 i 的接收消息、方法传播和返回结果.作为对象 i 的内部计算过程控制器,Console 可以通过 S/R 演算的 ICP 进程表达式进行细化,而细化的每一步都可以看做是对象 i 的交互计算行为的形式描述:

$Console=R(x:M).A\mid Console$(任意时刻接收请求消息)

$A=[\psi]B+[\neg\psi]B'$(ψ 用于检查请求消息类型,封装、动态绑定等)

$B=S(i;x_1,\rho_I(\partial_1(i,head(x_3))),\rho_A(\partial_1(i,head(x_3)));x_3,x_4).C$(方法传播)

$B'=S(x).0$(不执行不符合条件的消息)

$C=$For $(k=2,k\leqslant length(\partial(i,head(x_3))),k^{++})D;H$(方法传播)

$D=R(y).E$(接收分解动作的返回结果)

$E=[\zeta]F+[\neg\zeta]F'$(ζ 用于检查消息类型,确认所返回结果的标识和传播动作)

$F=[y_4\neq\text{“DECLINE”}]G+[y_4=\text{“DECLINE”}]G'$

$F'=S(y).0$

$G=S(i;y_1,\rho_I(\partial_k(i,head(x_3))),\rho_A(\partial_k(i,head(x_3)));y_3,y_4).0$

$G'=S(i;x_1,head(x_2),x_3,\text{“DECLINE”}).0$

$H=R(y).P$

$P=[\chi]S(i;x_1,head(x_1),x_3,y_4).0-[\neg\chi]S(y).0$ (χ 确认最后一个传播动作的返回)

条件 ψ,ζ 和 χ 根据对象的需求具体来确定.每个进程表达式后面的注释用于解释控制器的一般作用.非叶对象控制器控制对象方法的传播,For $(k=2,k\leqslant length(\partial(i,head(x_3))),k^{++})D;H$ 是循环执行方法$(i,head(x_3))$的 $length(\partial(i,head(x_3)))-1$ 个分解动作(方法传播)的缩写形式.对于不满足叶对象状态约束的请求,对象将返回一个特殊值“DECLINE”.

· 若 i 为叶对象标识,则 $MC[i]=Console@i$.

叶对象的计算控制器控制对象方法对对象状态的访问和修改,并将计算结果返回上层环境.可以通过 S/R 演算的内部 ICP 形式描述叶对象的计算过程控制器.

$Console=R(x).A$(接收请求消息)

$A=[\psi]B+[\neg\psi]B'$(ψ 用于检查消息封装等)

$B=r_i?(s).C$(读当前对象状态)

$B'=S(x).0\mid Console$(消息不符合 ψ 条件)

$C=[\varphi]D+[\neg\varphi]D'$($\varphi$ 用于检查请求方法的前条件)

$D=w_i!(head(x_3)(s,x_1,x_2)).S(i;x_1,head(x_1),x_3,head(x_3)(s,x_1,x_4)).Console$

$D'=S(i;x_1,head(x_1),x_3,\text{“DECLINE”}).Console$

其中 r_i,w_i 属于内部动作集合 Internal.

3.3 对象状态

我们以 S/R 演算的交互计算进程的形式引入对象状态.将对象状态看作内部交互计算进程,

既符合对象状态不断变化的自身特点,又分"层次"地抽象出对象系统中包含的要素,使得我们可以用更具一般性的抽象模型来统一地刻画各类对象系统.

令 $r_i, w_i \in Internal, s \in Sorts$,把对象 i 的状态形式描述为对象内部交互计算进程 $St[i](s)$:

- 若 i 为叶对象标识,则 $St[i](s)=r_i!(s).St[i](s)+w_i?(s').St[i](s')$;
- 若 i 为非叶对象标识,则 $St[i](s)=\prod_{j\in I_i} St[j](s)$.

3.4 交 互

给定对象系统基调 $\Sigma=(I,A,IA,AR,\partial)$,$\forall i\in I$,对象 i 与其环境 $Env[i]$ 的交互计算进程 obICP为

$$obICP[i]=MC[i]|St[i](s).$$

其中,$MC[i]$为 $Ob[i]$的计算过程控制器,$St[i](s)$为 $Ob[i]$的内部状态计算进程.

至此,我们已经完整地给出了对象系统与其环境交互下的动态行为变化描述.通过 S/R 演算的操作语义,$MC[i]$的计算流可以刻画出:对象输入不受局部的控制、输出及内部动作受局部约束、对象封装、操作(方法)的动态绑定、方法传播以及对象内部的并发行为等.

4 总 结

对象系统的行为变化是系统内、外因有机统一的具体表征.因此,本文设计的组合构造模型力图将对象系统的主要计算特征纳入统一的抽象基调之中,通过定义基调上的一组关系,把对象系统的层次结构和内部协同工作有机地结合在一起.在此基础上,本文给出一个区分动作类型的交互演算(S/R 演算),描述组合对象系统的动态行为和交互,较好地刻画出对象封装性、对象计算的事件驱动性、动态绑定以及对象系统内部的交互与并发等固有特征.本文强调的是,一个对象的行为不可能脱离其所处环境而孤立地改变,组成对象和组合对象的行为之间服从一定的关联约束.由于篇幅所限,我们没有讨论对象系统行为的互模拟,这是需要进一步研究的课题.

References:

[1] Walker, D. Process calculus and parallel object-oriented programming languages. Parallel Computers: Theory and Practice, 1995. 369~390.

[2] Talcott, C. L. Composable semantic models for actor theories. LNCS 1281, Berlin: Springer Verlay, 1997. 321~364.

[3] Nierstrasz, O. Towards an object calculus. LNCS 615, Berlin: Springer-Verlay, 1991. 1~20.

[4] Honda, K., Tokoro, M. An object calculus for asynchronous communication. In: Proceedings of the ECOOP'91. LNCS 512, Berlin: Springer-Verlay, 1991. 133~147.

[5] Fisher, K. A delegation based object calculus with subtyping. LNCS 965, Berlin: Springer Verlay, 1995.

[6] Lynch, N. A., Tuttle, M. R. Hierarchical correctness proofs for distributed algorithms. Technical Report, MIT/LCS/TR-387, Laboratory for Computer Science, Massachusetts Institute of Technology, 1987.

[7] Agha, G. Concurrent object-oriented programming. Communications of the ACM, 1990,33(9):125~141.

[8] Cheung, To-yat. A survey on the integration of object technology and formal methods. Technical Report, DCS-TR-98-02, Hong Kong, City University of Hong Kong, 1998.

[9] Ehrich, H.-D. Object specification. In: Krcowaki, H.-J., *et al.*, eds. IFIP 14.3 Volume on Foundations of System Specification. 1997.

[10] Huang, Tao, Feng, Yu-lin, Li, Jing. A formal semantic model for object. Journal of Software, 1995,6: 207~212(in Chinese).

[11] Huang, Tao, Qian, Jun, Zhou, Huan. Object calculus I. Journal of Software, 1999,10(9):931~940 (in Chinese).

[12] Huang, Tao, Qian, Jun, Wang, Xu. Object Calculus II. Journal of Software, 1999,10(9):941~951 (in Chinese).
[13] Milner, R. Concurrency and Communication. London: Prentice Hall, 1989.

附中文参考文献:

[10] 黄涛,冯玉琳,李京.对象形式语义模型.软件学报,1995,6:207~212.
[11] 黄涛,钱军,周桓.对象演算 I.软件学报,1999,10(9):931~940.
[12] 黄涛,钱军,王栩.对象演算 II.软件学报,1999,10(9):941~951.

Composition and Interactive Computation of Object Systems*

QIAN Jun, HUANG Tao, FENG Yu-lin

(*Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*);
(*Object Technology Center, Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*)
E mail: qianjun@softel.com.cn
http://www.iscas.ac.cn

Abstract: The computation of object systems is an open and dynamic computing process with concurrent interactions, and the behavior of object systems presents the unification between internal causes and external causes, so that the research of dynamic behavior of object systems should not be separated from the discussion of static construction of object system. This paper aims at the specification of dynamic behaviors and interactions of objects, while a formal model for composite object systems is constructed to consolidate the formal abstraction and conception framework throughout the whole paper at first. A sorted CCS-like calculus named S/R is built up to clarify the computation process of object systems and to characterize the inherent properties of encapsulation, dynamic binding, interaction and concurrency, etc. It is emphasized that the behavior of each object is restricted by its environment and the behavior of each component object should be consistent with the behavior of its composite object.

Key words: object system; component construction model; S/R calculus; concurrency; interaction

* Received October 28, 1999; accepted May 19, 2000
Supported by the National Grand Fundamental Research 973 Program of China under Grant No. G1998030404; the National Natural Science Foundation of China under Grant No. 69833030