第 7 卷第 3 期
1996 年 3 月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 7, No. 3
Mar. 1996

基于事件驱动的主动对象模型*

朱 冰 梅 宏 杨芙清

(北京大学计算机科学与技术系 北京 100871)

摘要 本文总结了现有被动对象模型存在的不足,探讨了对象的动态行为的抽象和描述问题,提出并建构了基于事件驱动的主动对象模型.在该模型中,引入了主动对象,主动对象能检测到外部发生的事件以决定自己的行为;引入主动对象类,以对主动对象进行抽象描述;还引入了运作对象,以协调其他主动对象的行为,一个运作对象还代表了一个可运行的系统,可以通过继承复用已开发的系统,增加了继承的力度.

关键词 对象,被动对象,主动对象,事件,数据抽象.

根据对象的动态行为特点,可以将对象分为被动对象(passive object)、主动对象(active object)和 agent.

被动对象当且仅当有消息发给它时才动作,没有接到消息时,处于不活动状态.被动对象模型是当前面向对象技术的主要依据和基础.但被动对象不便于并发系统的描述.

主动对象除响应发送给它的消息外,还可主动执行其操作.主动对象研究中,常见的作法是把对象当作通讯进程[1],或把对象的操作当作进程.另外,主动对象的行为可以通过状态转换规则来描述.[2]每个转换规则都是"条件—动作"形式.一个转换规则的执行是原子的;当条件部分的任何条件变量值改变时,转换规则就被激活,条件满足时就执行其动作.

有关 agent 和多 agent 系统的研究已成为分布式人工智能研究的一个十分活跃的领域,涉及的方面有:多 agent 的体系结构、多 agent 程序设计、多 agent 之间的合作和协商以及多 agent 的学习等.[3~9]agent 有 2 个特点,一是自主性,一是相互合作性.agent 作为独立的个体,能自学习、自增长,同时又可以和别的 agent 进行磋商、合作以完成任务.

参照已有对象模型的研究成果,针对对象的动态语义的抽象和描述,本文提出了一个具有主动行为特征的对象模型.该模型考虑了主动对象主动执行的特点;同时也参考了 agent 的某些特性,如 agent 自身能直接控制它的动作;agent 能受环境影响并通过自身动作来影响环境等等.笔者通过研究实时系统和分布式专家系统的控制方式,拓广了"事件"概念的内涵,设计了事件驱动的主动对象模型.在该模型中,构成系统的对象能检测到外部发生的事

* 本研究得到国家 863 高科技项目基金资助.作者**朱冰**,女,1967 年生,博士,主要研究领域为软件工程,面向对象技术,程序设计方法.**梅宏**,1963 年生,副教授,主要研究领域为软件工程,面向对象技术,新型程序设计语言.**杨芙清**,女,1932 年生,中国科学院院士,教授,博士导师,主要研究领域为软件工程,系统软件,面向对象技术.

本文通讯联系人:朱冰,北京 100871,北京大学计算机科学与技术系

本文 1995—08—30 收到

件,并由此决定自己的行为;对象的运行可能导致某事件或事件集的变化.并发对象的同步是由事件来控制的.

本文首先总结了被动对象、主动对象和 agent 的特点,介绍了本模型的基本情况.第1节给出了模型的基本思想、基本概念和主要特点.第2节从主动对象类、系统运行等方面进一步介绍了本模型的情况.

1 模型简介

1.1 模型的基本思想

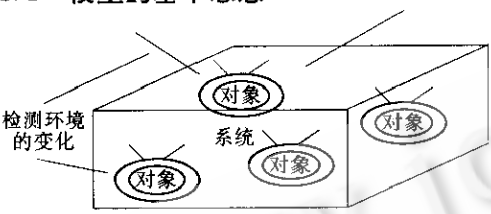

图1 主动对象模型示意图

软件系统的对象既有被动对象,也有主动对象.被动对象只能根据发送给它们的消息事件来完成有关的操作.而主动对象则具有主动行为能力,这些对象能了解周围环境的变化,并且能根据环境的变化自主地采取行动.反过来,对象的动作又有可能改变其周围的环境.本模型中,"环境"是借助于"事件"来表示的.事件的发生与否体现了环境的变化或改变,见图1.

1.2 模型的基本概念

(1) 主动对象

在本模型中,采用增强的数据抽象原则来设计软件系统的基本单位——主动对象.首先,根据数据抽象的原则来设计一组属性和作用在其上的操作:有哪些属性,其上操作的对外接口如何(即说明消息模式),操作是怎样实现的;其次,说明环境变化时,会运行哪些操作,也就是给出对操作的操纵说明信息;最后,将属性、作用在其上的操作、对操作的操纵封装在一个实体中,这一实体就被称为主动对象.

定义 1. 主动对象是软件系统的基本单位.由数据、作用于数据之上的操作、以及对这些操作的操纵说明组成.

定义 2. 运作对象是能协调其它主动对象行为的一种特殊的主动对象.

一个运作对象可以作为一个可运行的软件系统的代理,它的作用相当于C++程序的main()函数、Smalltalk 的消息序列以及 Eiffel 的根类对象.

(2) 主动对象类

抽象数据类型把一组数据与作用其上的操作组成一个实体,使得外界只知道它做什么,而不用知道它如何做,也不用知道其数据是如何表示的.这很适合于刻画被动对象.为了进一步明确地描述主动对象主动地与环境相互作用的能力,本文提出了主动对象类的概念.

定义 3. 主动对象类具有相同性质和行为的主动对象的集合.主动对象类为属于它的全部对象提供了统一的抽象描述.其抽象描述包括数据的描述、作用于数据之上的操作的描述以及对这些操作的操纵的描述.

(3) 事件

本模型是借助于事件使主动对象与其周围的环境发生联系.

定义 4. 事件是指主动对象外部发生的,主动对象感兴趣的所有活动、变化和事情.对一特定事件,它有3种状态:发生、未发生、情况不明.

事件具有发射能力,当某事件发生时,对象能捕获到它的发生.

在本模型中是把消息作为一类特殊事件来处理的.有一个消息发送给该对象,对对象而言就是发生了一个与之有关的事件.对于被动对象,消息是发生在它之外的全部事件;对于主动对象,消息只是它感兴趣的事件的一种.

1.3 模型的特点

基于事件的主动对象模型有如下特点:① 消息作为事件的一种,统一了消息和事件概念,减少了模型的基本概念,使得模型简单、明了.② 事件存在于对象之外,这就把对对象的控制与对象的属性分开,遵循了对象的封装性.③ 控制局部化,简化了系统控制的复杂程序,有助于并发系统、分布系统、以及智能系统的开发和研究.④ 既能描述主动对象,又能描述被动对象.通过将被动对象作为主动对象的退化形式而给出了对象概念的统一描述.

2 模型描述的说明

2.1 主动对象类的描述举例

考虑这样一个收容栈类.该收容栈类对象具有主动行为能力,能将系统中已老化(即暂时不用)的对象收集起来,而当系统要求再利用该收容栈类对象收集的老化对象时,则可以后进先出的方式获得一老化对象.下面给出收容栈类 *Stack* 的描述:

$$
\begin{aligned}
&Class\ Stack(Object)=Object+\\
&\quad S:stack\\
&\quad \Sigma:create:->stack\\
&\qquad push:stack\ object->stack\\
&\qquad pop:stack->stack\\
&\qquad read:stack->object\cup\{error\}\\
&\quad E:aging(obj)\\
&\qquad reuse\\
&\quad \Pi:register(aging(obj),push)\\
&\qquad leaving(reuse,pop)\\
&\quad A:pop(create)=create\\
&\qquad pop(push(s,n))=s\\
&\qquad read(push(s,n))=n\\
&\qquad \eta^{aging}\rightarrow(register(aging(obj),push)\wedge\bigcirc\rho^{push})\\
&\qquad \eta^{reuse}\rightarrow(leaving(reuse,pop)\wedge\bigcirc\rho^{pop})
\end{aligned}
$$

其中 S 是类集,其元素与类型 *stack* 有关.Σ 是操作符号的集合,说明一组对象上的操作的定义情况,有创建 *create*、压入 *push*、弹出 *pop* 和读栈 *read* 操作.E 是事件集,有对象老化事件和要求再利用收容栈中对象事件.Π 是操纵说明集,说明操作的运用情况,*register* 指当系统中发生对象老化事件时,收容栈对象将老化对象压入栈中;*leaving* 指当系统要求再利用老化对象时,收容栈对象提供一个老化对象.A 用于描述对象特性,给出操作的性质和事件与操作的联系,其中 η 是事件发生变量,记载事件集中事件的发生与否;ρ 是操作运行变量,记载操作的运用情况.如 $\eta^{aging}\rightarrow(register(aging(obj),push)\wedge\bigcirc\rho^{push})$ 表示当系统中有 *aging* 事件发生后,收容栈对象下一时间就将老化对象压入栈中.

2.2 系统运行的简要描述

通过时序变量 η(与事件有关)和 θ(与对象有关)来描述.

(1) 假定系统的对象是 $o_1,o_2,\cdots,o_n$.

(2) 系统外部发生的事件序列为 $\{e_1,e_2,\cdots,e_m\}$,对外部事件暂仅考虑顺序出现情况.并且不妨将该序列作为系统运行时间的划分依据.

(3) 引入描述系统运行时环境的时序变量 η 和 θ.

① η 记载系统运行过程中所有可能涉及的事件的发生情况,是一个多元组.

$$\eta=(\eta^1,\eta^2,\cdots,\eta^m,\cdots)$$

对每一项变量 η^i,在 j 时刻可能取值为 0,1,⊥.0 表示事件 e_i 在 j 时刻没发生;1 表示事件 e_i 在 j 时刻发生;⊥表示事件 e_i 在 j 时刻发生与否还不明确.

有函数 $happen(e)$,表示事件 e 当前发生,要修改时序变量 η 的值.

$\eta[happen(e)]$表示改变项变量 η^i 的当前时刻取值为 1.

② θ 记载系统运行过程中所有可能涉及的对象运行与否的情况,也是一个多元组.

$$\theta=(\theta^{o_1},\theta^{o_2},\cdots,\theta^{o_n},\cdots)$$

对每一项变量 θ^{o_i},在 j 时刻可能取值为 0,1,⊥.0 表示对象 o_i在 j 时刻没运行;1 表示对象 o_i 在 j 时刻在运行;⊥表示对象 o_i 在 j 时刻运行情况不明确.

有函数 $add(o)$,表示对象 o 当前开始执行,要修改时序变量 θ 的值.

$\theta[add(o)]$表示改变项变量 θ^o 的当前时刻取值为 1.

(4) 根据主动对象类的情况,我们不妨假定对象如果开始运行,它可根据 η 和 θ 的当前取值确定自己的具体操作.

(5) 对系统的运行情况,我们假定可由 η 和 θ 确定的状态空间来说明.由此下面采用数学归纳法来描述 η 和 θ 的取值.

① $\eta_0^1=1 \quad \eta_0^i=0 \quad i=2,\cdots,m \quad \eta_0^j=\perp \quad j>m$

$\eta_0=(1,0,\cdots,0,\perp,\cdots)$

$\theta_0^{o_i}=0 \quad i=1,\cdots,m \quad \theta_0^{o_j}=\perp \quad j>m$

$\theta_0=(0,\cdots,0,\perp,\cdots)$

② $\bigcirc^{i+1}\theta=\bigcirc^i\theta[add(o_j)]$ 其中 o_j 是集合 A_i 中元素

$A_i=(ran\ D_{oo})\bigcup(ran\ (D_{ce}\circ(\omega_{[1=e_i]}D_{co})))$

$\bigcirc^{i+1}\eta=\bigcirc^i\eta[happen(e_k)]$ 其中 e_k 是集合 B_i 中元素

$B_i=\{e_i\}\bigcup(ran\ (\omega_{[(1=o_j)\wedge\theta_i^j]}D_{oe}))$

其中 D_{co}表示类和对象之间的分类关系;D_{ce}描述类和事件之间的关系;D_{eo}表示事件对象之间的事件运行关系(事件的出现会导致某些对象的运行和变化);D_{oe}表示事件对象之间的事件出现关系(对象的运行,会导致某些事件的出现).对关系 R,F,G,$ran\ R$ 是求 R 的值域;$F\cdot G$是 G 对 F 的右复合;$\omega_{[\omega]}R$ 是 R 的选择 ω 运算.

3 结 语

本文提出并建造了一种基于事件驱动的主动对象模型.现在,我们已完成模型的建造工

作,并且在基于事件的主动对象模型的基础上,结合 BDOL 语言[10]统一形式的描述框架,设计了 BDOL+语言.[11]该语言既能描述对象的主动行为特征,又能复用已有 OOPL 定义的混合类.另外,已完成了 BDOL+语言编译器和 BDOL+库管理系统的设计.[11]BDOL+语言系统的实现正在进行之中.

参考文献

1 Nierstrasz Oscar. Regular types for active objects. ACM OOPLSLA'93, 1993. 1～15.

2 Minoura T, Pargaonkar S, Rehfuss K. Structural active object systems of simulation. ACM OOPSLA'93, 1993. 338～355.

3 Kraus Sarit. Agents contracting tasks in non—collaborative environments. Proceedings. of 11th National Conference on AI, 1993. 243～248.

4 Alterman R, Zito—wolf R. Agents, habitats, and routine behavior. Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993. 305～310.

5 Weish Gerhard. Learning to coordinate actions in multi—agent systems. Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993. 311～315.

6 Schwuttke U M, Quan A G. Enhancing performance of cooperating agents in real—time diagnostic systems. Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993. 332～337.

7 Henz Martin, Smolka Gert, Wurtz Jorg. Oz—a programming language for multi—agent systems. Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993. 404～409.

8 Zlotkin Gilad, Rosenschein Jeffrey S. A domain theory for task oriented negotiation. Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993. 416～422.

9 Ephrati Eithan, Rosenschein Jeffrey S. Multi—agent planning as a dynamic search for social consensus. Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993. 423～429.

10 杨芙清,梅宏,朱冰.BDOL 语言及其支撑系统.电子学报,1996,(2).

11 朱冰.软件复用技术研究[博士论文].北京大学,1994.

ACTIVE OBJECTS MODEL WITH EVENTS

Zhu Bing　　Mei Hong　　Yang Fuqing

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

Abstract　In order to solve the under generalization to object dynamic behaviors in passive objects models, this paper presents an active objects model which abstracts active behaviors of objects. In this active objects model with events, an active object could probe whether an event happens outside, and then automatic determines its own action. Active objects class is discussed, which could give abstract description of those active objects having the same properties. Computing objects are introduced to coordinate other objects. A computing object might be an agent of a running system.

Key words　Object, passive object, active object, event, data abstract.