491-494

1998年 8月 北京航空航天大学学报 August 1998
第24卷 第4期 Journal of Beijing University of Aeronautics and Astronautics Vol.24 No.4

(29)

一种基于面向对象Petri网的并发程序建模方法[1)]

任爱华 牛锦中 张永鸣
(北京航空航天大学 计算机科学与工程系)

TP311.5

摘 要 介绍了一种基于面向对象Petri网的并发系统建模方法.该方法把面向对象技术与Petri网理论相结合,构成一种面向对象Petri网,可以解决用Petri网建立并发程序模型所遇到的状态爆炸问题,又使得建模系统具有可重用性且易于维护,是一种具有数学和图形方式相结合的形式化描述.

关键词 软件工程;软件工具;建立模型;Petri网;面向对象技术;并发系统建模

分类号 TP 311.5

面向对象是一种非常有效的程序设计范型,越来越受到人们的重视和欢迎[1].这是因为用此种方法开发出来的应用系统能适应不断变化的客观环境的需要,并可在构件级上实现软件重用.但是在并发程序的设计中,运用面向对象方法构造复杂大系统时,往往也要构造大量的对象实体,它们之间的关系也变得越来越复杂,且诸多对象之间隐含有并发性,对象之间相互通讯与制约关系的表达不直观,稍一疏忽便易出错,因而迫切需要有一种更加直观的、形式化的方法来进行辅助设计.

而Petri网能简洁地描述系统的动态特性(如:并发、同步、冲突等)和系统中的资源及约束条件,并有相应的分析方法(如:可达图分析、不变量分析以及图论等相关的分析理论),而且是一种图形工具,易于理解和描述,所以被广泛地应用在具有并发、并行、异步和随机性质的系统建模与分析中.然而传统的Petri网模型,对大系统来说存在模型的复杂性问题.尽管可通过采用高级Petri网和化简方法来降低模型的复杂性,但效果并不十分理想,实现较为困难.

如果采用面向对象的Petri网模型则比以往传统的Petri网模型描述更直观,因为面向对象技术提供了抽象的封装、分类以及继承机制,所以对庞大而复杂的系统描述提供了更有效的简化手段.

由以上可知,将面向对象技术与Petri网方法相结合,充分发挥二者的优势,同时尽量弥补各自的缺点,无疑是一种解决复杂问题的根本方法[2].

1 面向对象Petri网的形式化描述与系统建模

面向对象技术对解决复杂性问题以及可重用问题提供了有效的方法.而Petri网适合于描述系统的动态特性和系统中的资源约束条件,既有形式化描述,又可采用图形直观地表示.在面向对象技术中,把对象看为实体,每个对象有自己的数据和任务,根据接收到的通讯信息(消息)来完成相应的任务.为了使每个对象尽可能相互独立,采用了将同步制约条件从每个对象内部分离出来的做法,使得对象内部行为描述与外部通讯接口的分析可分开处理,从而提高可重用性.

下面分别来讨论在面向对象Petri网技术中对象的描述、网的建立、通讯同步约束的提取以及系统分析的解决办法.

1.1 对象的表示

考虑一个对象*A*,如图1所示.它由基本对象*AA*和*AB*复合而成.图2给出了*AA*和*AB*的细节.在图中,用框子圈住对象的内部,表示封装与抽象,对象的外部接口部分由"消息队"(用椭圆表示,其作用类似于用圆表示的库所place)、"门"(用粗线表示,其作用类似于用方形框表示的变迁transition)以及它们之间的流关系(用弧线表示)给出.各对象内部分别可以有若干实体(用黑点表示,即标志token),各token所处的place表示各自的当前状态.

本文所介绍的面向对象Petri网简称为OPN,在

收稿日期:1998-05-12 第一作者 女 40岁 副教授 100083 北京
1)航空科学基金(96F51075)资助项目

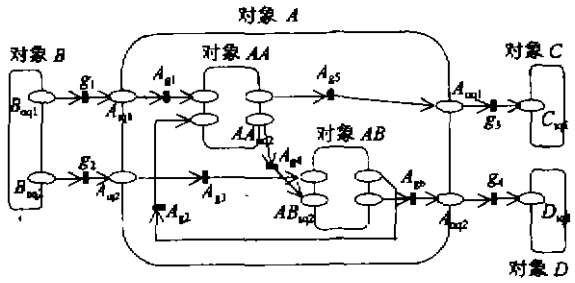


图1 复合对象 A 及对象 B、C、D 构成的系统

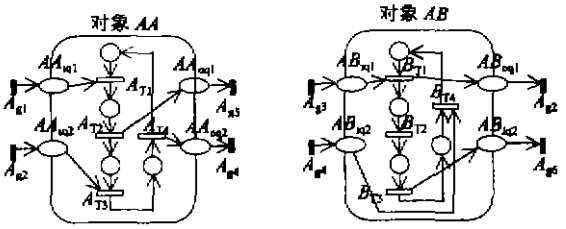


图2 对象 A 的内部结构:基本对象 AA 和 AB

OPN 中定义了2种对象类型:基本对象与复合对象.在基本对象中,不包含并发部分,基本对象只用于表示顺序行为与静态性质;而在复合对象中则允许并发,因为复合对象由具有独立行为的基本对象构造而成,其控制分布在各基本对象中,并且根据系统要求可以来同步这些基本对象的顺序行为.

1.2 OPN 的形式化描述

1) 系统

在面向对象 Petri 网中,系统由多重层次对象及它们之间的消息传递关系组成:

$$S = (O, R)$$

其中 O 为对象集合;R 为关系集合.

例如:图1所示系统:

$O = \{A, B, C, D\}$;

$R = \{\langle B_{oq1}, g_1, A_{iq1}\rangle, \langle B_{oq2}, g_2, A_{iq2}\rangle, \langle A_{oq1}, g_3, C_{iq1}\rangle, \langle A_{oq2}, g_4, C_{iq2}\rangle\}$.

2) 对象的外部结构

对象 $O_i \in O$ 可表示为六元组:

$$O_i = (H_i, G_{ii}, G_{oi}, M_{ii}, M_{oi}, F_i)$$

其中 H_i 为对象层次,指定父对象;G_{ii} 为输入门集合;G_{oi} 为输出门集合;M_{ii} 为输入消息队集合;M_{oi} 为输出消息队集合;F_i 为流关系集合.

输入/输出门统称为门,用来连接相应的输出消息队和输入消息队,并进行必要的消息类型转换.

输入/输出消息队是对象的窗口,通过它们可以向外部发送消息要求服务并接收应答或从外部接收消息提供服务并返回应答.这样的消息传递机制有效地将对象的内部与外部相分离,提高了可维护性和可重用性.

流关系集合指示输入门与输入消息队、输出门与输出消息队的连接关系.

3) 对象的内部结构

设 O_{pi} 表示基本对象 i,O_{cj} 表示复合对象 j,则整个系统的对象集合为

$$O = O_p \cup O_c$$

其中 $O_p = \bigcup_i O_{pi}$;$O_c = \bigcup_j O_{cj}$.

进一步,用 I_{poi} 和 I_{coj} 分别表示 O_{pi} 和 O_{cj} 的内部结构.

关于复合对象,其内部结构定义了该对象所包含的子对象以及它们之间的相互关系:

$$I_{coj} = (X, Y, R_j)$$

其中 $X \in \rho(O_c)$;$O_{cj} \notin X$;$Y \in \rho(O_p)$;R_j 表示关系集合.

这里的 $\rho(O_c)$、$\rho(O_p)$ 分别表示 O_c、O_p 的幂集.比如,图1中的 A 对象的内部结构:

$$I_{Aco} = (\varnothing, \{AA, AB\}, \{\langle AA_{oq2}, A_{g6}, AB_{iq2}\rangle\})$$

关于基本对象,其内部结构明确指定了其静态特性与动态行为.静态特性是用代数方法来描述对象的某些属性或状态;而动态行为则用高级网来表示,因而具有更强的表达分析能力.

基本对象的内部结构可定义为

$$I_{poi} = (D_i, V_i, S_i, T_i, F_{li}, N_i, M_0)$$

其中 D_i 为属性集合;V_i 为状态变量集合;S_i 为状态集合;T_i 为活动 transition 集合;F_{li} 为局部流关系集合;N_i 为实例集合;M_0 为初始标识.

属性与状态变量意思上很相似,都表示基本对象的当前状态,但前者常表示较长久的量(如人的年龄等);而后者则常随实例状态的变化而变化(如机床的忙、闲).用 place 代表基本对象的当前状态,所以状态是 place 的非空子集.

每个状态都对应有刻划状态变量的一元状态谓词,定义状态谓词是通过将特定状态映射为基本对象的状态值元组的一个函数来完成.

活动 transition(即 action transition)是 transition 的子集,起着同步作用,在它的先决条件满足时,就可以引发,去执行预先定义的活动.这些活动代表了顺序程序的执行.活动分为内部活动和外部活动,主要取决于该活动是否为其它对象提供了服务.

局部流关系表示了基本对象的内部控制流,指示输入/输出消息队(或状态)与活动 transition

之间的连接关系.

模型中有2种token:一种代表对象的具体实例,只在对象内部流动,不允许在网执行期间被创建或销毁.实例由实例标识符唯一确定和引用.实例的初始状态由初始时存放实例的place表示;另一种token代表对象之间通讯的信息,可在对象之间流动,允许动态创建或销毁.

1.3 利用OPN建立系统模型

对于一个应用系统,建立面向对象的Petri网模型分以下几步:

1) 确定组成系统的各个对象的结构.

① 从研究的问题域里出现的名词中选出用来构成系统的对象.

② 对每一对象确定其自身行为及与其它对象的联系.

③ 对较复杂的对象,可进一步提取子对象,对每一子对象进行②中的分析;对已比较简单的对象,用简单Petri网构造出该对象的内部行为.

④ 用消息队表示对象的外部接口.

⑤ 将内部控制中的transition与相应的消息队相连.

2) 构成系统静态结构.

① 确定对象间的通讯关系.

② 用门将对象间对应的输入消息队与输出消息队相连.

3) 确定系统的初始标识.

① 确定各对象的实例.

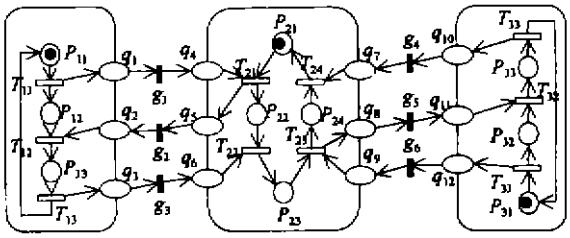
② 确定各实例的初始状态.

③ 将代表实例的各个token放入与各自初始状态对应的place中.

经过以上3步,就可以建立一个面向对象的Petri网系统模型.图3给出了生产者/消费者问题的面向对象Petri网模型,其中包含1个生产者、1个消费者和1个容量为1的缓冲区.生产者(消费者)在进行生产(消费)之前,须先通过g_1门(g_6门)向缓冲区提出请求,待应答消息通过g_2门(g_5门)返回许可之后,才能开始生产(消费).结束时,还要通过g_3门(g_4门)告诉缓冲区.很显然,图3清楚地表示了系统涉及的生产者、消费者、缓冲区3个对象的结构及相互通讯关系,充分体现了面向对象的思想及Petri网图形化的优点.

1.4 同步约束的提取及系统分析

建立了系统的Petri网模型之后,必须进行各种性能分析以确定该系统模型是否可靠及符合实际,这其中最重要的是死锁检测.



生产者　　缓冲区　　消费者

P_{11}—生产者空闲;　T_{11}—提出生产请求;
P_{12}—等待生产许可;　T_{12}—获得生产许可;
P_{13}—正在生产;　T_{13}—结束生产;
P_{21}—缓冲区为空;　T_{21}—收到生产请求并应答;
P_{22}—等待生产结束;　T_{22}—缓冲区变满;
P_{23}—缓冲区为满;　T_{23}—收到消费请求并应答;
P_{24}—等待消费结束;　T_{24}—缓冲区变空;
P_{31}—消费者空闲;　T_{31}—提出消费请求;
P_{32}—等待消费许可;　T_{32}—获得消费许可;
P_{33}—正在消费;　T_{33}—结束消费

图3　生产者/消费者问题模型

面向对象Petri网的死锁检测过程是:首先根据对象的内部结构,提取出对其输入/输出门发生次序的要求,构造出接口等价网(Interface Equivalent Net,简称IE网),然后将不同对象的IE网合并,构成整个系统的IE网,通过建立IE网的可达树,分析其中是否存在死锁.

具体分为以下几步:

1) 局部分析

① 对每一基本对象的内部行为进行可达树分析.

② 利用可达树中各transition的发生次序来确定通过消息队与各transition相联系的输入/输出门的发生次序.如:与生产者相连接的几个门的引发序列可由如下方法获得:

· g_1　通过q_1与T_{11}相连;

· g_2　通过q_2与T_{12}相连;

· g_3　通过q_3与T_{13}相连.

③ 根据门的发生次序构造IE网.

2) 同步分析

① 将各IE网中出现的相同的门合并,得到对象间的IE网.

② 对得到的IE网进行可达树分析.

3) 层次抽象分析

① 通过内层的IE网确定与内层各门相连的外层复合对象的各门的发生次序,构造出IE网.

② 对得到的IE网进行可达树分析.

在上面3步中,若在某处发现有死锁,则建立的模型需要改进以消除死锁;否则,说明无死锁.

以图3中的生产者/消费者系统为例进行分析，图4～图6分别描述了该系统中的3个对象——生产者、消费者、缓冲区的IE网的构造过程，得到的整个系统的IE网与缓冲区的IE网恰好相同，显然该网无死锁.

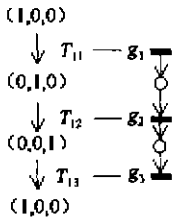


图4 生产者对象的可达树与IE网

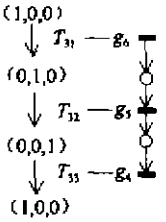


图5 消费者对象的可达树与IE网

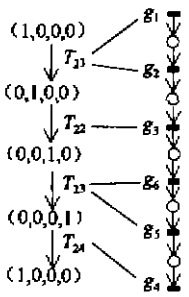


图6 缓冲区对象的可达树与IE网

2 结束语

本文讨论的面向对象Petri网模型有以下优点：

1）对象的内部结构通常较简单，容易识别；同时对象通过消息来请求或提供服务，从而体现了对象的外在作用，这样就使一个复杂的系统变得容易理解.

2）对象的内部结构与外部接口相分离，提高了可维护性.

3）以对象作为构成系统的元素，从而可以分步分层地进行分析和通讯检测，简化了系统的验证，而且出现的不一致性可方便地定位.

4）每一对象是一类事物的模板，可简单地增加或减少token数来改变实例数目，为受资源限制的情况提供了一种实用模式.

总之，可维护性与可重用性是面向对象Petri网的最大优点.虽然因增加门与消息队而使transition与place数目有所增加，但利大于弊.

另外，面向对象Petri网中尚未加入时间约束条件.若能扩展成时间网(timed Petri nets)，将会使建模能力进一步增强.

与面向对象技术相结合的Petri网建模方法种类很多，但大多数只适用于专门的应用.本文介绍的面向对象Petri网建模方法力图在一般性的并发软件开发上提供一条新的思路.目前正着手于这方面的研究与开发工作.

参考文献

1 蔡希尧，陈 平.面向对象技术.西安：西安电子科技大学出版社，1995

2 Lee Y K, Park S J. OPNets: an object-oriented high-level Petri net model for real-time system modeling. J Systems Software, 1993, 20(1):69～86

Object-Oriented Petri Net Based Method for the Concurrent Program Modeling

Ren Aihua Niu Jinzhong Zhang Yongming

(Beijing University of Aeronautics and Astronautics, Dept. of Computer Science and Engineering)

Abstract An approach for concurrent system modeling based on the object-oriented high-level Petri net is described. The O-O Petri net is constructed by combining object-oriented techniques with Petri net theory. The modeling of the O-O Petri net not only can reduce the complexity of the model, but also make the modeled system with reusability and easy maintainability. This approach makes O-O system modeling with the description for both formalization and graphics. As an illustration, the producer and consumer concurrent pattern is modeled using object-oriented Petri nets. The modeling experience with these nets demonstrates that the decoupling and separation of knowledge and constraints clearly enhance maintenance and reusability in concurrent system modeling.

Key words software engineering; software tools; model building; Petri nets; object-oriented techniques; concurrent system modeling