

基于UML顺序图的Petri网建模

杨玉梅^{1,2},刁永锋¹

(1.西华师范大学 计算机学院,四川 南充 637000;

2.绵阳师范学院 计算机科学与工程系,四川 绵阳 621000)

摘要:UML(统一建模语言)是一种功能较强的面向对象图形建模工具,但其过程描述中缺乏严格的语义,而Petri网不仅有形象直观的图形描述还有严格的数学定义。针对一个简单运输系统的建模将二者结合起来,阐述了UML类图、顺序图的Petri网建模转换方法,避免直接进行Petri网建模过程的烦琐,使建模过程清晰可见。

关键词:UML;Petri网;建模

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2007)10-0130-04

Petri Net Modeling Based on UML Sequence Diagram

YANG Yu-mei^{1,2}, DIAO Yong-feng¹

(1. Computer College of China West Normal University, Nanchong 637000, China;

2. Computer Science and Project Department of Mianyang Normal University, Mianyang 621000, China)

Abstract:UML is a strong function of the object-oriented graphical modeling tool, but it's lack of strict semantic description of the process, and Petri nets not only have the visual image of the graphic description, but also have strict mathematical definitions. In this paper, a simple modeling of the transport system will allow the two to work together, expounded the UML class diagram, sequence diagram modeling ways to avoid the cumbersome process of direct Petri net modeling, the modeling process is clearly discernible.

Key words:UML;Petri net;modeling

0 引言

传统的系统分析和设计存在许多不足,比如开发出来的软件与实际需求不符、维护困难、软件复用率低等。为了解决这一系列的问题提高软件设计质量,人们提出了面向对象的方法,而面向对象的分析与设计应该从建立模型开始。模型的建立是对现实系统的一种抽象,它的建立反映现实系统的特点,同时又区别于现实系统。能否高效率高质量地建立系统模型,建模工具的选择是关键。文中实例中涉及UML建模的类图和顺序图,其中类图体现各对象之间的静态关系,顺序图描述各对象之间在时间顺序上的交互关系,这些特点恰好是Petri网建模所关心的要素,结合UML建模和Petri网的建模方法,实现了从UML到Petri网的建模。

1 UML

统一建模语言(UML)统一了Booch,Rumbaugh和Jacobson的表示方法,并对其作了进一步的改进,最终成为能为大众所接受的标准建模语言。UML是一种图形化建模语言,因为结合了软件工程领域面向对象的设计思想,因此具有定义良好、易于表示、表现直观等特点,可谓是建模的好帮手。目前,UML已经成为建模语言的工业标准^[1],应用领域包括软件系统和非软件系统领域。但是UML作为一种图形建模语言语义不严格只能半形式化地对系统进行描述。UML的建模提供了9种图,它们分别是用案图、类图、对象图、状态图、活动图、顺序图、协作图、构件图、部署图。这些图反映了不同的系统元素,元素之间的联系构成了系统,其中元素之间的联系是通过关系来实现的,这些关系也是模型元素^[1],常见的关系有关联、聚合、组合、泛化、依赖、实现。

2 Petri网

Petri网是1962年由德国科学家C. A. Petri先生在其博士论文《用自动机通信》中首次提出的,是对离

收稿日期:2006-12-04

基金项目:四川省科技厅重点科研项目(05JY029-093)

作者简介:杨玉梅(1978-),女,四川仁寿人,硕士研究生,主要研究方向为Petri网、计算机应用;刁永锋,教授,硕士研究生导师,研究方向为计算机辅助教育应用。

散并行系统的数学表示,适合于描述异步的、并发的系统模型。Petri 网有严格的数学定义和直观的图形表达方式,具有相对成熟的语义和可执行性^[2],系统结构分析技术比较完善,可以对系统的活性、可达性、有界性等进行有效验证。所以在进行系统模型的分析 and 设计过程中可以将直观易懂的 UML 图和具有严格数学定义的 Petri 网二者有机结合起来使用。

Petri 网是一个二元关系有向图,其中包括库所和变迁两类节点。库所(图中用小圆圈表示)代表系统中表示状态的节点,其中状态信息用托肯(token)(小圆圈里的小黑点)表示;变迁(图中的小矩形块)是系统中活动的部分。

2.1 Petri 网基本定义^[3]

定义 1 三元组 $N = (P, T; F)$ 称作有向网的充分必要条件是:

- (1) $P \cap T = \emptyset$;
- (2) $P \cup T \neq \emptyset$;
- (3) $F \subseteq P \times T \cup T \times P$ (“ \times ”为笛卡尔积);
- (4) $\text{dom}(F) \cup \text{cod}(F) = P \cup T$, 其中

$$\text{dom}(F) = \{x \mid \exists y: (x, y) \in F\}$$

$$\text{cod}(F) = \{y \mid \exists x: (x, y) \in F\}$$

定义 2 令 $N = (P, T; F)$ 为网, $X = P \cup T$ 为其元素集, 设 $x \in X$ 为 N 的任一元素,

(1) $x^- = \{x \mid (y, x) \in F\}$ 称为 x 的前集(pre-set)或输入集。

(2) $x^+ = \{z \mid (x, z) \in F\}$ 称为 x 的后集(post-set)或输出集。

定义 3 六元组 $\Sigma = (P, T; F, K, W, M_0)$ 为网系统的条件是:

- (1) $N = (P, T; F)$ 构成有向网, 称为 Σ 的基网。
- (2) K, W, M_0 依次为 N 上的容量函数、权函数和标识。 M_0 称作 Σ 的初始标识。 $M_0: P \rightarrow \{1, 2, 3, \dots\}$ 。

定义 4 变迁发生条件。

变迁的触动必须在标识满足条件的情况下才能进行。设 M 为网系统 Σ 的基本网上的任一标识, $t \in T$ 为任一变迁

(1) $t^- = t^- \cup t^+$ 称为 t 的外延。

(2) t 在 M 有发生权的条件是:

$$\forall p \in t^-: M(p) \geq W(p, t) \wedge \forall p \in t^+: M(p) + W(p, t) \leq K(p)$$

t 在 M 有发生权记作 $M[t >]$ 。

定义 5 变迁发生结果。

如果 $M[t >]$, 则 t 在 M 可以发生, 将标识 M 改变为 M 的后继 M' , M' 的定义如下:

对 $\forall p \in P, M' =$

$$\begin{cases} M(p) - W(p, t) & \text{若 } p \in t^- - t^+ \\ M(p) + W(t, p) & \text{若 } p \in t^+ - t^- \\ M(p) - W(p, t) + W(t, p) & \text{若 } p \in t^- \cap t^+ \\ M(p) & \text{若 } p \notin t^- \end{cases}$$

M' 为 M 的后继记作 $M[t > M']$ 。

2.2 Petri 网相关性质^[3]

(以下提到的 Petri 网都是四元组^[4], 其中略去了权函数和容量函数。)

定义 6 冲突关系。

设 $\Sigma = (P, T; F, M_0)$ 为一个 Petri 网, M 为网的任一标识, $t_1, t_2 \in T$ 为 Σ 任意两个变迁, 若 $M[t_1 >] \wedge M[t_2 >]$, 但是 $\neg M[\{t_1, t_2\} >]$, 则 t_1, t_2 在 M 冲突。

定义 7 并发关系。

设 $\Sigma = (P, T; F, M_0)$ 为一个 Petri 网, M 为网的任一标识, t_1 和 t_2 在 M 并发的充分必要条件是:

$$t_1^- \cap t_2^- = \emptyset \wedge t_1^+ \cup t_2^+ \subseteq M$$

定义 8 可达性。

Σ 网系统的可达集 $[M_0 >]$ 是满足下列条件的最小集合:

- (1) $M_0 \in [M_0 >]$;
- (2) 若有 $M' \in [M_0 >]$, $t \in T$, 使 $M'[t > M]$, 则 $M \in [M_0 >]$ 。

定义 9 Petri 网的有界性、安全性和活性。

(1) 若对于所有 $M \in [M_0 >]$, 存在正整数 k , 使得对所有 $p \in P, M(p) \leq k$, 就说 Σ 是有界的, 当 $k = 1$ 时称 Σ 是安全系统。

(2) 对 $t \in T$, 若对任一可达标识 $M \in [M_0 >]$, 均有从 M 可达的标识 $M' \in [M >]$, 使得 $M'[t >]$, 就说变迁 t 是活的。

(3) 若所有 $t \in T$ 都是活的, 就说 Σ 是活的。

在实际的 Petri 网建模的过程中, 要充分利用 Petri 网的这些严格的定义为依据建立和验证系统模型。

3 UML 和 Petri 网的建模应用

这里以一个简单的运货机工作过程为例来进行 UML 和 Petri 的建模。运货系统主要包括控制台、左右活动杆、传送器、机车。前提假定所有设备能正常运行, 整个流程描述如下:

(1) 控制台初始化;

(2) 控制台同时启动两个活动杆;

(3) 两个活动杆将货物抬起放入传送器, 传送器再将货物运入机车, 一次传送完毕控制台从第一步开始继续传送, 系统如图 1 所示。

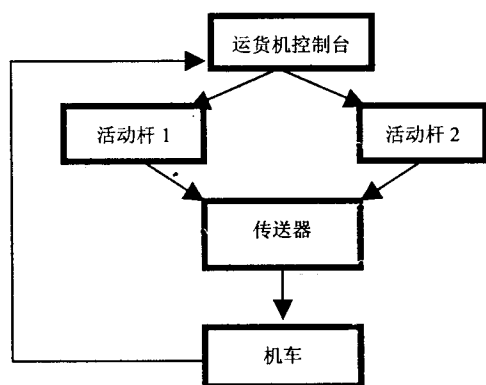


图 1 运货机工作流程图

3.1 类图和顺序图的建立

类图是对一组具有相同属性、动作等对象(类的具体实例)的总体描述。类图能够比较直观地描述各对象之间的关系,便于分析和设计系统,也为动态模型的建立奠定了基础。类图用矩形表示,包括类名、属性和动作,根据情况类图中属性和动作可以省略。对图 1 的货物传送系统的 UML 类图建模如图 2 所示。

类图是一种静态图,因此系统具体的运行情况在类图中并不能得到充分体现,要体现系统的运行情况应该采用动态建模,动态建模在 UML 中有多种,在这里采用顺序图进行动态建模。顺序图采用的二维图关系主要体现对象之间基于时间的交互关系,在这个二维图中横向代表系统中所涉及的对象,纵向代表时间,对象的生命周期由其下方的生命线表示,对象之间通过传递消息来进行交互,生命线之间的箭头表示消息的传递,表示方法如图 3 所示^[1]。通过顺序图可以很清晰地看到各对象之间执行的先后关系及交互关系。图 4 为系统的顺序图建模,在该图中只用到了同步箭头和表示并行的箭头(图中的“启动

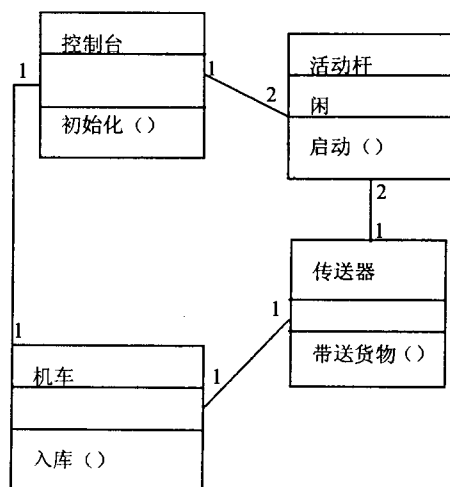


图 2 货物传送系统的 UML 类图

(())”部分),其中还用到了消息的嵌套,即对象自身引用自身的消息,在图中体现为系统的“初始化”。

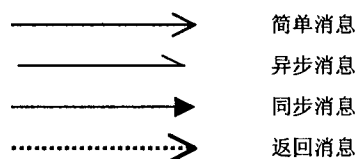


图 3 消息的表示法

3.2 从 UML 顺序图到 Petri 网的转换

正如前面所提到的 UML 是一种半形式化的建模工具,通过建立顺序图很清晰地看到整个系统运行过程,但是还不能充分地验证系统的有效性。Petri 网具有严格的数学定义和有效的验证方法,因此将 UML 转换成 Petri 网以便验证运货系统模型的有效性,在此针对图 4 进行转换。

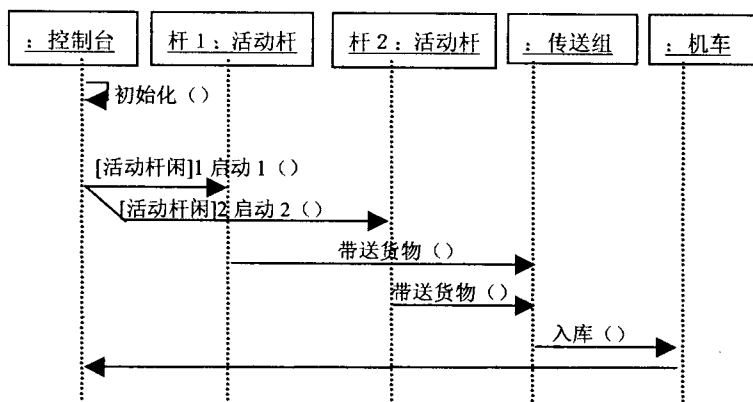


图 4 货物传送系统的 UML 顺序图

在针对文中实例进行转化的过程中应遵循以下几个规则:

(1) 将顺序图中的消息“启动()”、“带送货物()”和“入库()”转换成一个基本的顺序 Petri 网,如图 5 中虚框内的结构所示。p2 到 p6 和 p3 到 p7 分别是启动活动杆 1 和活动功杆 2 的过程,其中 p17 和 p18 是作为条件存在。p8 到 p12 和 p9 到 p13 是带动货物的过程。p14 到 p16 是入库的过程。每个动作都有开始、处理和结束三个状态,状态之间由变迁过渡^[5]。

(2) 如果是嵌套调用动作,即对象自身调用自身的动作,处理方法跟(1)一样,不过这里简化成一个库所表示,即就是一个开始状态。如图中 p1。

(3) 消息之间的转变(或者是传递)由一个变迁表示,如图中 t1, t4, t6, t14, t17。

(4) 循环系统的起止状态用同一个库所表示。

有时候所建立的 Petri 网模型外观上看上去非常复杂,这样可读性差,分析的时候也容易出错,为了能够看上去更简洁且保持系统的原本性质,根据 Petri 化简方法^[6],将图 5 最终化简为如图 6 所示,按理还可以将图 6 进一步简化,但是为了能更清楚地了解系统的

运行过程就化简到此。根据前面的定义 8 和定义 9 验证该 Petri 网是可达的、活的并且是有界的,所以这是一个有效的 Petri 网模型。

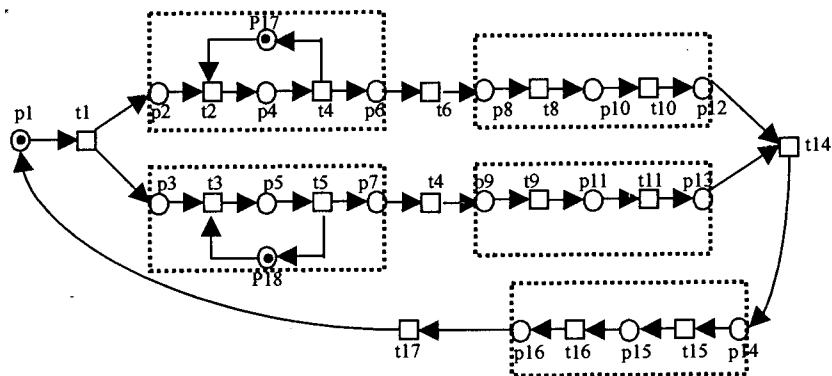


图 5 货物运送系统的 Petri 网模型

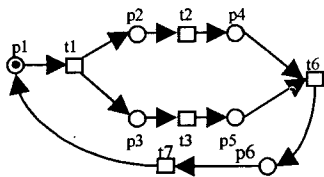


图 6 货物运送系统的 Petri 网简化模型

4 结束语

文中通过对一个简单的货物运输系统模型的分析

与设计,充分利用了 UML 的直观、易懂的性质,分别用类图和顺序图描述了系统对象的结构组成以及它们之间的动态行为关系,并且在此基础上结合了 Petri

网的具有严格数学定义和有效的验证方法的特点,将 UML 和 Petri 网有机结合起来,实现了 UML 顺序图到 Petri 网的转化。通过实践证明这种转化是有效的和可利用的。

参考文献:

- [1] 徐宝文,周毓敏,卢红敏. UML 与软件建模[M]. 北京:清华大学出版社,2006:55-63;85-92.
- [2] 汪文元,沙基昌,谭东风. 基于 Petri 网的 UML 活动图工作流程建模比较[J]. 系统仿真学报,2006(2):504-510.
- [3] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社,2005:18-23;35-37.
- [4] 柯飞帆,宁宜熙. 基于 Petri 网和 UML 的工作流模型设计[J]. 南京航空航天大学学报,2006(2):121-125.
- [5] 周长红. UML 的 Petri 网建模[D]. 青岛:山东科技大学,2004:52-60.
- [6] 吴哲辉. Petri 网导论[M]. 北京:机械工业出版社,2006:167-169.

(上接第 129 页)

- (3) 如果 $v > n$ 或 $\text{flag} = 0$, 则转(9);
- (4) 如果栈 stack 是空栈, 则将第 w 个入栈序列符号入栈, $w++$;
- (5) 如果 stack 栈顶元素值等于序列的第 v 个元素, 或者 $w \leq n$ 则转(7);
- (6) 第 w 个入栈元素压入 stack 栈中, $w++$, 转(5);
- (7) 如果 stack 栈顶元素值等于序列的第 v 个元素, 则匹配, stack 栈顶元素出栈, $v++$; 否则, 说明所判断的序列不是出栈序列, 赋 flag 为零;
- (8) 转(3);
- (9) 结束, 若此时 flag 为 1, 则序列为出栈序列, 否则不是。

将该算法替代算法 1 中的相应部分, 称为改进后的算法 1。从表 1 可以看出改进后的算法 1, 在运行时间上比原算法大大缩短, 甚至比算法 2 还要好。

4 结束语

由前置 O 栈序列可构造出其对应的二叉树, 用该二叉树表示入栈和出栈操作比较直观, 易于理解; 文中

给出三种算法, 实现对 n 个元素依次入栈, 求其全部出的出栈序列, 并对这三种算法进行了分析和研究。由于当 n 很大时, 求出所有出栈序列是一个不现实的问题, 所以如何判断一个序列是否为出栈序列显得更为重要, 文中结合栈的性质给出一种时间复杂度为 $O(n)$ 的判断某一序列是否为出栈序列的算法, 提高了程序的执行效率。下一步, 将继续对栈的性质进行研究, 探讨栈在其他领域中的应用。

参考文献:

- [1] 卢开澄. 组合数学[M]. 第 2 版. 北京:清华大学出版社, 1991:119-130.
- [2] 徐凤生. 出栈序列的性质及其求解新算法[J]. 计算机工程与应用, 2006, 42(5):66-68.
- [3] 严蔚敏, 吴伟民. 数据结构(C语言版)[M]. 北京:清华大学出版社, 1997:152-155.
- [4] 唐保祥. 栈序列及其生成算法[J]. 郑州大学学报:自然科学版, 2001, 33(4):33-35.
- [5] 范年柏, 张大方, 颜学义, 等. 基于栈操作的用例规模的一个计算公式[J]. 湖南大学学报:自然科学版, 2004, 31(6):80-82.