

基于分层 Petri 网的仓储管理建模与验证

赵 珊¹, 马光思¹, 毛宏燕²

(1. 西安建筑科技大学 信息与控制工程学院, 陕西 西安 710055;

2. 华东师范大学 软件学院, 上海 200062)

摘 要:文中综合理论研究和仓储管理项目的维护经验。应用 Petri 网理论, 介绍了 Petri 网的存储机制, 并提出了基于工作流和 Petri 网分层建模的实施策略、操作步骤和一套简明案例。详细介绍并给出了采用自动工具 WoPeD2.0 对部分建模实例的验证分析和修改过程, 分析讨论了 Petri 网的存储机制及在模型验证中的支柱作用。经分析验证, 得出实验结果正确无误。研究工作重点是提供一套基于 Petri 网建模分析的实施步骤和简明案例, 旨在推动这一形式化方法在工程领域的广泛应用。

关键词:分层 Petri 网; 仓储管理; 工作流; 存储机制

中图分类号: TP393.0

文献标识码: A

文章编号: 1673-629X(2010)12-0029-05

Modeling and Verification of Inventory Management Based on Layered Petri Net

ZHAO Shan¹, MA Guang-si¹, MAO Hong-yan²

(1. School of Info. & Con., Xi'an Univ. of Arch. & Tech., Xi'an 710055, China;

2. Software Engineering Institute, East China Normal University, Shanghai 200062, China)

Abstract: Combined with theoretical research and inventory management project maintenance experience. Applied Petri nets theory, the implementation strategies and operation procedures are proposed based on workflow and the Petri nets layered model and a set of brief case are given. The validation analysis and modification process of modeling samples are introduced with automatic tools WoPeD2.0. The storage mechanism of Petri net and its pillaring effect in model validation are discussed. Through analysis and experimental results verify correct. This research focus provides implementation steps and a set of brief case based on the Petri nets layered model, to promote the formal methods widely used in engineering field.

Key words: layered Petri net; inventory management; work flow; storage mechanism

0 引 言

鉴于有可视化自动分析和验证工具的支持, 比起其他形式化方法, 基于 Petri 网的应用系统建模分析更为流行。

工程应用中, 配合工作流技术和 UML 的 Petri 网模型验证等方面的研究工作成绩显著^[1-4]。辅助 Petri 网自动模型分析工具验证、更新设计, 对理清系统逻辑, 确保模型正确和加速实际项目维护进度, 都有重要意义。

1 建模策略、实施步骤与仿真

基于 Petri 网的模型分析与验证, 是对传统软件方法和手段的补充与增强。某些常用的优秀方法和策略, 如对问题求解的逐步细化和分层思想, 可以移植到基于 Petri 网的模型构建中。其中, 建模策略归纳如下:

(1) 结合工作流方法和遗留文档, 分层建模原则保持自顶向下, 实施灵活处理;

(2) 由简到难, 以训练和掌握应用建模步骤、约定规则和主体框架为主;

(3) 逐步开展系统存储机制、功能逻辑及工作原理等关键技术的研究。

1.1 顶层功能建模与仿真

顶层模型针对整个仓储管理系统。

Step1: 检查原系统设计方案和文档, 补充完善原

收稿日期: 2010-03-18; 修回日期: 2010-06-15

基金项目: 国家自然科学基金(90718004); 陕西省教育专项基金(07JK306)

作者简介: 赵 珊(1984-), 男, 硕士研究生, 研究方向为计算机软件与理论; 马光思, 教授, 研究方向为软件方法、信息安全。

始文档。原始仓储管理系统划分的相关子系统有:进货、收货、摆放、出库等。

Step2:若无描述 Step1 关于功能划分的模块关系图,按照 workflow 技术,补充文档如图 1 所示。其中,放置在方框中的模块代表几个子系统,有向弧表示各模块之间的作业流向关系。

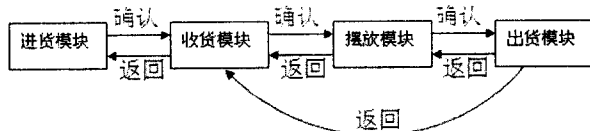


图 1 仓储管理系统的功能模块关系

Step3.1:令图 1 每个模块对应于 Petri 网中一个用小圆圈表示的库所,并给每个库所编号;具体对应关系为:进货模块对应 p1,收货模块对应 p2,摆放模块对应 p3,出库模块对应 p4;

Step3.2:将图 1 中关联于模块之间的每条有向弧,一分为二(保持与原弧方向一致),在分断处插入一个用方框表示的变迁,给各变迁编号并指定其具体含义。

Step4:仿真分析。

顶层 Petri 网模型与仿真效果见图 2。其中,各编号变迁的具体含义指定为:t1 为运送;t2 为合格性检验,不合格的被退回;t3 为分类、甄选;t4 为因无法分类或甄选不合格而退回;t5 为被售出、被借出;t6、t7 为返还或退货等。

仿真前仅 p1 放有一个 token,图 2 中黑点为 token 运行到 p2,阴影框为授权变迁。

仿真过程仅一个库所含有 token,其他库所都是空的,即模型符合有界性:

$$\forall p \in P, M \in [M_0], M(p) \leq K(p)$$

其中, K 为给定容量函数。

对初始标识 $M_0[1,0,0,0]$,启动发生变迁,生成的所有可达标识从可达图可见,该模型是有界的、没有死锁。亦即,顶层模型有效、可达。

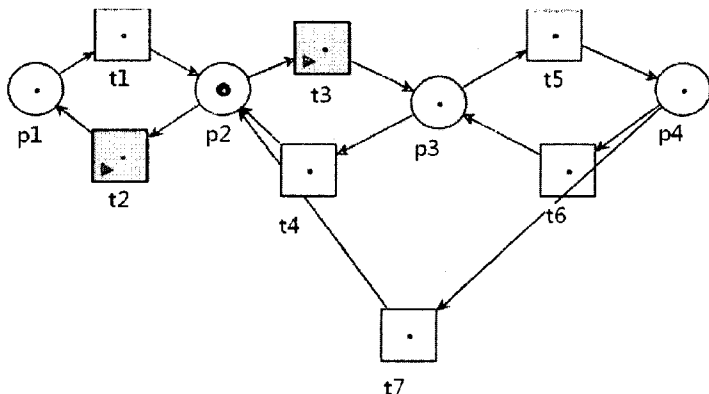


图 2 仓储管理顶层 Petri 网模型

若在 p1 中放多个 token,如初始标识为 $M_0[2,0,0,0]$,也可验证模型的可达性和有界性。但验证结果的可达标识多达 10 个。

1.2 分层子网建模与仿真

对子网的建模与仿真可多人分工进行。注意,1.1 节各步骤所含规则仍然适用。

Step1:找出子模块对应的库所和涉及的变迁,考察是否需要细化并建立分层子 Petri 网模型。例如,按 workflow 技术,收货模块对应的 p2 可细化为计划收货、收货处理及收货检查三个作业步骤:

Step1.1:根据前一步进货模块发来的进货清单和仓库当前状态,对统计待收货物品,验收等工作制定接收计划;

Step1.2:收货处理,将待收货物按制定好的计划完成接收、确认等工作;

Step1.3:检查接收情况,对出现问题的检查结果,及时返回前一步,或给出退货建议。建立收货模块的基本作业流程。

Step2:根据作业流程建立收货模块的子网模型。

建立收货模块的基本作业流程和 Petri 网子网模型如图 3 所示。

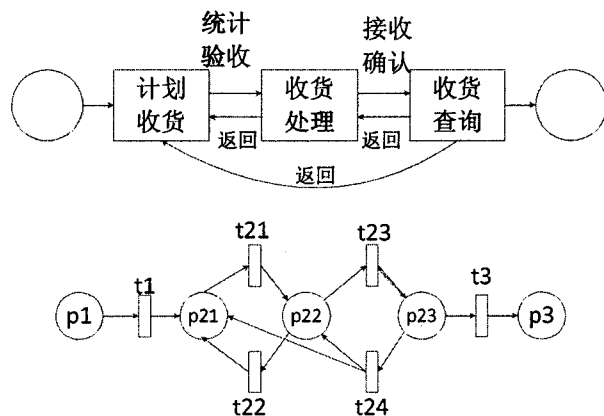


图 3 收货模块的基本作业流程和收货模块子网模型(框中)

图 3 中, p21 为计划收货, p22 为收货处理, p23 为收货检查; t21 为统计验收, t23 为接收确认, t22, t24 和 t25 为返回上一步或返回第一步。

同理,摆放模块对应的 p3,可以细分为选择货仓、选择货区及确定货位三个步骤完成。对应的作业流程与建立的摆放模块的子 Petri 网模型与仿真如图 4 所示。

其中, p31 为货仓, p32 为货区, p33 为货位; t31 为确认货区, t32 为确认货

位, t_{33} 为返回。

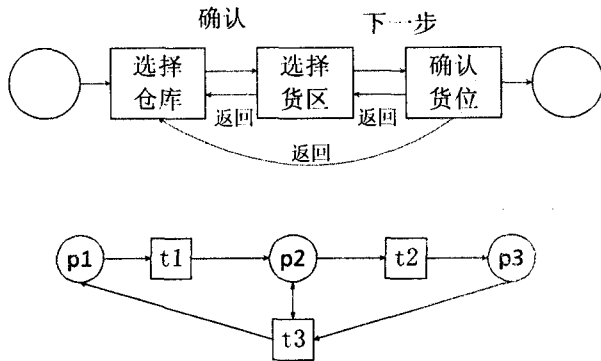


图4 摆放模块基本作业流程与
摆放模块子网仿真过程

建模阶段及时发现并排除冲突的意义很大。因为凭视觉直观审查模型往往很难查出有些弊病,拖到代码调试阶段检查,就要付出更多精力和时间。

2 存储机制及其支柱作用

研究存储机制,能加深对 Petri 网的深层了解,对正确理解自动验证工具显示形式、功能原理,应对建模分析中出现的各种问题有积极辅助作用。

形式化定义的 Petri 网基网,可具体用集合、序偶结构、图结构和关联矩阵等来存储。研究这些存储机制,能加深对自动验证工具的深层了解,对正确应对构建分析模型中出现的各种问题有潜在的辅助作用。

2.1 Petri 网基本理论

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。

Petri 网基本定义:

第一,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:

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

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

第二,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) $\cdot x = \{x \mid \exists y: (y, x) \in F\}$ 称为 x 的前集(pre-set)或输入集。

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

第三,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、权函数和标志。 M_0 称作 Σ 的初始标志。 $M_0: P \rightarrow \{1, 2, 3, \dots\}$ 。

第四,定义4:变迁发生的条件。变迁的触动必须在标志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志, $t \in T$ 为任一变迁:

(1) $\cdot t = \cdot t \cup t$ 成为 t 的外延。

(2) t 在 M 有发生权的条件是:

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

t 在 M 有发生权记作 $M[t >]$ 。

第五,定义5:变迁发生的结果。如果 $M[t >]$, 则 t 在 M 可以发生,将标志 M 改变为 M 的后继 M' , M' 的定义如下:

对 $p \in P, M' =$

$M(p) + W(p, t)$ 若 $p \in \cdot t - t$ 。

$M(p) - W(p, t)$ 若 $p \in t \cdot - t$ 。

$M(p) - W(p, t) + W(p, t)$ 若 $p \in \cdot t \cap t \cdot$ 。

$M(p)$ 若 $p \notin \cdot t \cup t \cdot$ 。

M' 为 M 的后继记作 $M[t > M']$ 。

Petri 网相比有限状态自动机,增加了活动部分,也就是增加了过程。用变迁表示。通常 Petri 网有向图来描述,用圆圈或椭圆表示库所,用矩形表示变迁,用从 x 到 y 的箭头表示流关系中的 (x, y) 。

2.2 结点、序偶

形式化定义的3元组 Petri 网基网 $N = \langle P, T, F \rangle$, 有限库所集 $P = \emptyset$ 和有限变迁集 $T = \emptyset$ 均可用集合表示。定义过程包括给出库所及变迁的标识符 id , 即给出库所及变迁的命名列表。详细地,还包括库所及变迁在图示中的坐标位置。通过指定源和目的结点描述反映二元关系的弧集 $F \subseteq (S \times T) \cup (T \times S)$, 可采用序偶,或在结点的属性域给出指针,利用链表和图的形式定义。格式可为 XML、文本等。

参见图4,摆放模块子网的集合、序偶结构,其数学形式为:

$P = \{p_{31}, p_{32}, p_{33}\}; T = \{t_{31}, t_{32}, t_{33}\};$

$F = \{ \langle p_{31}, t_{31} \rangle, \langle p_{32}, t_{32} \rangle, \langle p_{32}, t_{33} \rangle, \langle p_{33}, t_{33} \rangle, \langle t_{31}, p_{32} \rangle, \langle t_{32}, p_{33} \rangle, \langle t_{33}, t_{31} \rangle, \langle t_{33}, p_{32} \rangle \}$

这种形式可用 XML、文本或其他格式描述:

(1) 独立定义库所和变迁集 P 及 T 。定义过程包括给出库所及变迁的标识符 id , 即给出库所及变迁的命名列表。详细地,还有库所及变迁在图示中的坐标位

置。

(2) 通过指出源和目的结点描述弧集 F 的 8 个序偶对应的弧。

(1) 及 (2) 的 XML 格式的例句为:

```
< place id = "p31" > "中心点坐标位" < /place
>
< transition id = "t31" > "中心点坐标位" < /
transition >
< arc source = "p31" target = "t31" > "斜率或曲率"
< /arc >
```

2.3 图的形式

用图的形式存储基网 $N = \langle P, T, F \rangle$ 需要数据结构方面的知识^[5], 包括定义结点, 在属性域指明结点类型(库所/变迁)及坐标位, 声明指针。这可用数组、邻接表或十字链表等实现, 进而再用 XML、文本或其他格式描述。图 3, 4 仅仅给出了页面显示效果, 对应地, 系统内部应该有相关的数据存储。采用可视化工具, 跟踪选择、移动及最后所确定库所、变迁及弧线图标的页面操作, 对应结点的位置和属性, 弧线的源、目的和方向即可确定。同时, 也产生了相应的存储机制。反之, 若用代码给出存储机制, 相应地, 执行代码就会导出图示形式。解析一种存储机制不难导出其他存储和显示形式。例如, 可以根据集合、序偶存储导出关联矩阵, 展示页面图形; 根据关联矩阵可以导出集合、序偶存储机制, 展示页面图形; 也可记录页面图形的构造过程生成序偶存储机制, 并导出关联矩阵。

2.4 矩阵与向量

Petri 网的基网 $N = \langle P, T, F \rangle$ 可以表示为关联矩阵^[6,7], 使得矩阵的行对应库所, 列对应变迁。对此, 通常先定义输入矩阵 W^- ($W_{ij}^- = 1$, 当且仅当 $p_i \rightarrow t_j \in F$) 和输出矩阵 W^+ ($W_{ij}^+ = 1$, 当且仅当 $t_j \rightarrow p_i \in F$)。易见, 矩阵 W^- 对应着所有从库所到变迁的有向弧; W^+ 对应着所有从变迁到库所的有向弧。由 $W = W^+ - W^-$ 即得关联矩阵。

Petri 网的标识 M (Mark) 为库所集 P 到集合 $\{0, 1, 2, \dots\}$ 的映射, 由处于库所 p_i 中代表资源的 token 数目 $M(p_i)$, $i = 1, 2, \dots, n$ 惟一确定, 通常定义为向量序列。组合 $N = \langle P, T, F \rangle$ 和初始标识 M_0 , 形成了以 4 元组 $\Sigma = \langle P, T, F, M_0 \rangle$ 定义的网系统。网系统中, 随着 token 移动可以产生多个标识。一个标识代表网的一种状态, 所有标识组成的状态图中, M_0 是初始状态, 也称状态图的根结点。

Petri 网的运行反映为网中 token 的流动, 这可导致状态标识 M 的变更。token 的流动依据 Petri 网的启动机制, 即针对某一状态标识 M , 检测输入及输出条件,

看是否可以触发启动某个或某些变迁。当触发可启动(被授权)变迁 tk 时, 通过改变输入、输出库所中的 token 数以产生新标识 M^* 。特殊记法为 $M[tk > M^*$ 。

从初始状态 M_0 出发, 通过激发某变迁序列 $t_1 t_2 \dots t_n$ ($\sigma = M_0 t_1 M_1 t_2 M_2 \dots t_n M_n$ 是 Σ 的一个有限出现序列, 当且仅当 $\forall i, 1 \leq i \leq n; M_{i-1}[t_i > M_i$; σ 的长度 $|\sigma| = n$) 可导出若干新标识。

2.5 求网系统的所有可达状态标识

(1) 对已有标识 M 求新标识 M^* 的算法。

Step1 计数器 i 置 1; 授权列表 List_ T 置空;

Step2 当 $(i \leq \text{变迁数 Count})$ 执行

Step2.1 若 (变迁 t_i 满足触发条件) 则

变迁 t_i 并入 List_ T ;

Step2.2 $i = i + 1$;

Step3 若 List_ T 为空, 则退出;

Step4 En_ Count 置授权变迁数; 计数器 i 置 1;

Step5 当 $(i \leq \text{授权变迁数 En_ Count})$ 执行

Step5.1 若 (变迁 t_i 不与其他变迁冲突) 则触发变迁 t_i ;

Step5.2 若 (存在冲突变迁) 则

根据增加互补条件的规则消除冲突;

Step6 结束。

(2) 利用关联矩阵求状态标识。

根据 Petri 网理论, 隐式 $M[tk > M^*$ 可用关联矩阵表示为显式: $M^* = M + WS^T$, 其中, S 是变迁序列的特征向量。对 $t_i \in T$, S 的第 i 个分量等于在序列中出现的次数, $i = 1, 2, \dots, n$ 。例如: 变迁序列 $t_1 t_1 t_3$ 的特征向量 $S = (2, 0, 1, 0, \dots, 0)$ 。

对图 4 收货模块子网, 令 t_{21}, t_{23} 发生, 即有变迁序列 $t_{21} t_{23}$, 则 $S = (1, 0, 1, 0, 0)$, 设 $M_0 = (1, 0, 0)$, 由 $M^* = M + WS$, 得

$$W \cdot S = \begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & -1 \end{bmatrix} \cdot (1 \ 0 \ 1 \ 0 \ 0)^T = (-1, 0, 1)$$

$$M^* = M_0 + WS^T = (1, 0, 0) + (-1, 0, 1) = (0, 0, 1)$$

3 仿真与验证示例

实验 1 对顶层 Petri 网模型仿真^[8]: 为验证可达性, 设 P_1 只有一个 token, 对初始标识 $M_0[1, 0, 0, 0]$, 启动发生变迁, 所生成的所有可达标识见图 5^[9]。

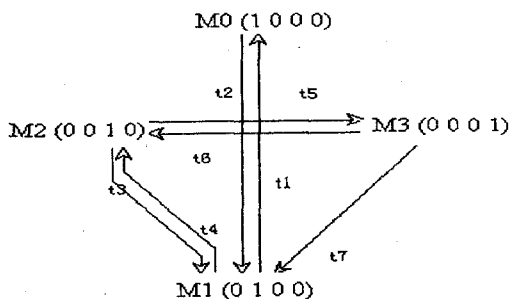


图5 仿真顶层 Petri 网可达图

在实验 1 的仿真过程中仅一个库所始终含有 token,其他库所都是空的,此系统符合有界性($\forall p \in P, \forall M \in [M_0, M(p) \leq K(p)$ 注: K 为给定容量函数)。从初始状态总能通过激发某变迁序列来激发任意变迁,而且从可达图可以看出,该模型是有界的、没有死锁,结论是,仓储管理系统的顶层模型有效、可达。

在 $P1$ 中放入两个 token,每个 token 也可自如运行,并且能够到达任意库所,符合可达性。

实验 2 对图 4 摆放模块分析,出现了故障:token 停止在最右端的库所处不能继续运行,没有被授权的变迁出现^[10]。

出故障的原因是,构建过程有违顶层模块中 Step3.2 约定的变迁插入规则,把表面上的几个相似返回混为一谈。按 workflow 技术,返回有可能回到前一步,也有可能回到其前各步。向不同位置的返回所携带的信息有一定差异,因差异不大而将几个模块压缩在一起是程序员的潜在习惯,由此引起的控制耦合常产生不稳定代码段。对模型进行修改,为摆放模块子网添加可用变迁 $t4$ 和 $t5$,更新后的结果如图 6 的虚线框所示。

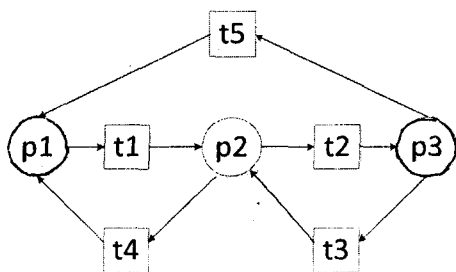


图6 更新后摆放模块的子网模型

仿真验证修改后的摆放模块子网,token 的运转流畅,可到达任意各个库所,没有遇到任何的障碍,修改成功,完成了预期设计。

实验 3 将顶层网模型与收货子网模型相结合对

综合模型仿真^[11]:

在实验 3 的仿真过程中,token 的运转流畅,可到达任意各个库所,没有遇到任何的障碍,证明了将分层子网和顶层网综合到一起也是可达、有效的。

4 结束语

文中介绍了采用 WoPeD2.0 自动工具对仓储管理维护项目进行建模分析与验证的方法和过程。遗产系统文档不规范、所用工具不完善和维护经验不足,都给工程实践带来了许多困难和挑战。论文的研究工作重点集中在提供一套基于 Petri 网建模分析的实施步骤和简明案例,旨在推动这一形式化方法在工程领域的广泛应用。有关复杂应用及所涉及特殊 Petri 网系统^[12]研究的关键技术和成果,将另文讨论。

参考文献:

- [1] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社, 2005.
- [2] 钟 越. 扩展 Petri 网在工作流建模中的研究与应用[D]. 西安:西安建筑科技大学, 2007.
- [3] 程念胜, 张宜生, 李德群. 一种基于 Petri 网的工作流元模型研究[J]. 计算机工程与应用, 2008, 44(24): 2-3.
- [4] 宋瑜辉. 基于 UML 和 Petri 网建模的研究与应用[D]. 西安:西安建筑科技大学, 2009.
- [5] 蒋昌俊. Petri 网的行为理论及其应用[M]. 北京:高等教育出版社, 2003.
- [6] 吴振寰, 王鹏伟. Petri 网关联矩阵与 PNML 描述之间的转换[J]. 计算机工程与应用, 2006, 42(21): 2-3.
- [7] 郑继川, 胡正国. 基于 Petri 网的工作流建模与分析[J]. 计算机工程, 2003, 29(5): 26-27.
- [8] 潘德桥. 基于 UML-Petri 网的舰载作战系统信息流建模与分析[D]. 武汉:华中科技大学, 2006.
- [9] 周建涛, 叶新铭. 一种构造 Petri 网可达图的方法[J]. 内蒙古大学学报(自然科学版), 1999, 30(3): 390-392.
- [10] 谢彦辉, 姚淑珍, 郭 峰. 顺序图至 Petri 网转化方法的研究与实现[J]. 计算机工程, 2006, 32(6): 260-262.
- [11] Bresi L. Some Preliminary Hints on Formalizing UML with Object Petri Nets[C]//Proceedings of the 6th world conference on Integrated Design and Process Technology, IDPT-2002. Pasadena, USA: [s. n.], 2002: 3-6.
- [12] Girault C, Valk R. Petri Nets for Systems Engineering: a Guide to Modeling, Verification and Application[M]. Berlin Heidelberg: Springer-Verlag, 2003.

中国计算机学会会刊、中国科技核心期刊
《计算机技术与发展》欢迎订阅,邮发代号:52-127