

# 层次结构的进程网

郭峰 乔磊 毛文祥

(北方工业大学计算机学院 北京 100144)

**摘要** 进程网(Process Net)是一种结合进程代数和Petri网理论的Petri网模型。当系统过于复杂时,进程网系统模型在实际应用中会遇到节点爆炸的问题,此时需要引入层次结构。提出了层次结构的进程网,给出了层次进程的建模过程和实现算法,解决了大的复杂系统建模和状态空间爆炸问题,可以清楚地反映出模型的层次,便于对进程网进行细化得到准确的模型,也便于用逐步求精、自顶向下的方法对被模拟系统进行建模,帮助用户实现各种粒度的仿真服务。

**关键词** 进程网,层次结构,细化,二叉树

**中图分类号** TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.11.015

## Hierarchy Structure of Process Net

GUO Feng QIAO Lei MAO Wen-xiang

(School of Computer, North China University of Technology, Beijing 100144, China)

**Abstract** Process net is a novel petri net model with a combination of process algebra and petri net theory. Process net system model is used in practical applications, however, when the system is too complicated, node explosion problem will be encountered, at this moment, it is necessary to introduce process net with hierarchy. A process net with hierarchical structure was presented and the hierarchical model and modeling algorithm were given. It solves the problem of large, complex systems modeling and settles space explosion. It can clearly reflect the level of model, so as to facilitate the process net refinement to obtain the accurate model. It also makes it easy to use stepwise refinement, top-down approach to complete modeling of the simulate system and help users to achieve a variety of size simulation services.

**Keywords** Process net, Hierarchy structure, Refinement, Binary tree

## 1 序言

Petri网<sup>[1]</sup>是常用的系统仿真工具,它可以描述系统的结构和模拟系统的运行。在描述服务组合时Petri网有自身的局限性,缺少对服务间接口信息的描述,无法精确分析服务组合对服务内部行为的影响<sup>[2]</sup>。本文将分析一个新的建模工具——进程网,其为一种结合进程代数和Petri网理论的Petri网模型。

文献<sup>[3]</sup>介绍了Petri网的分层模型,其为用Petri网对复杂系统建模的一种手段和方法<sup>[4]</sup>。在Petri网的分层模型<sup>[5]</sup>中,Petri网中的某些基本元素(库所或变迁)又可以细化成一个Petri网。

进程网最初是由基本进程网组合而成。层次结构的进程网使进程网的各个组成部分之间具有层次关系,保存进程网的层次关系可更好地分析系统的组成,对复杂流程进行仿真时,需要用到多层次进程网。

本文提出了进程网层次的概念、层次进程网模型、层次进

程网模型的建立过程,并实现了进程网分层。研究层次结构的进程网,对规模较大的系统的仿真和分析有着重要的作用,并方便进程网进行细化。

本文第2节介绍了进程网的几个基本概念;第3节提出了层次进程网模型;第4节介绍了层次结构进程网的建立过程;第5节介绍了进程网分层的实现;第6节是实验结果分析;最后总结全文。

## 2 进程网的几个基本概念及特点

**定义 1**<sup>[6]</sup> 进程网(Process Net, PrN)是一个七元组 $(P, T, F, A, \beta, I, O)$ ,其中 $(P, T, F)$ 是一个Petri网; $A$ 是动作集合,包含3种动作形式, $a$ 表示内部动作, $!a$ 表示发送消息动作, $?a$ 表示接收消息动作; $I \in T$ ,表示接收消息的变迁集合, $O \in T$ 表示输出消息的变迁集合, $I$ 和 $O$ 是PrN的接口变迁集合。 $\beta$ 是标记函数, $(T) \rightarrow A \cup \{\beta\}$ ,每个变迁用动作进行标识。 $I$ 中变迁以 $?a$ 形式标记, $O$ 中变迁以 $!a$ 形式标记<sup>[2]</sup>。进程网可以是基本进程网,也可以是基本进程网按照组合运

到稿日期:2015-08-05 返修日期:2016-02-14 本文受国家自然科学基金(61070030,61370051),北京市教委人才创新团队计划(4062012),2015年北京市大学生科学研究与创业计划项目成果资助。

郭峰(1972-),男,博士,讲师,主要研究方向为软件工程、软件测试与检验、Petri网、面向服务计算等,E-mail: guofeng\_buaa@163.com;

乔磊(1987-),男,硕士,主要研究方向为软件工程、Petri网等,E-mail: qiaoleioffice@163.com;毛文祥(1994-),男,主要研究方向为软件

工程、软件开发技术,E-mail: 2210302909@qq.com.

算规则进行组合运算之后生成的进程网。

PrN 综合了工作流网<sup>[7]</sup>和开放网<sup>[8]</sup>的特点,与工作流网相比,进程网新增两个概念:接口变迁和变迁标记;同开放网相比,进程网用变迁表示接口,同时进程网只有一个开始库所与一个终止库所,这些特征使得 PrN 具有 Petri 网图形化表示、进程代数的语法形式和遗忘式的结构化操作语义。

由系统论的分析可知,任何大系统都具有层次结构<sup>[9]</sup>。由于进程网没有分层的概念,导致进程网模型的逻辑不清晰,当系统过于庞大时,不方便建模。

在进程网中引入分层,即对组成进程网的基本组件和复杂组件集合进行划分。当系统过于繁琐时,通过分层可以看到系统的层次结构。

称较为抽象的规范到更具体的规范的转换步骤为细化<sup>[10]</sup>。进程网的细化指在进行自顶向下建模时,某一部分进程网还不能详细地描述该部分的业务流程,需要对其进行更加详细的分析和仿真。进程网的细化需要遵行对应的细化规则。

二叉树是每个节点最多只有两个子树的结构,子树有左右次序之分,次序不能颠倒。进程网的细化过程中,每次进行组合运算都是两部分进行,因此可以使用二叉树来实现进程网的分层。

### 3 层次进程网模型

#### 3.1 层次进程网模型分析

将复杂、大型的工作流的设计分为两个阶段:1)在全部工作流结构的顶层定义任务;2)在较低层次上确定任务的具体描述<sup>[11]</sup>。3层结构的 Petri 网模型如图 1 所示。

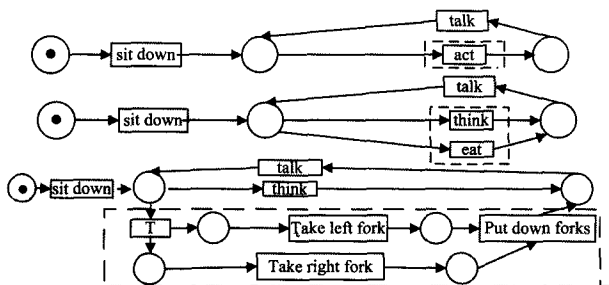


图 1 三层结构的 Petri 网

根据软件工程在设计时的自顶向下、逐步求精的思想引出层次进程网模型。当对一个系统进行建模与仿真时,先分析上层进程网的模型,然后再针对某一部分的进程进行细化分层,从而建立层次进程网,可以方便地看出进程网的组织结构和各种状态变化。

层次进程网的递归定义为: $MLPRN = \{TOP, S, PRN, T, K, R\}$ 。 $MLPRN$  不包括进程网的最基本组件的定义,它的定义包含在  $PRN$  中。其中,用  $TOP$  来标识进程网模型的父  $MLPRN$ , $TOP$  为 0 标识本模型为顶层模型, $TOP$  不为 0,进程网有子  $MLPRN$  模型; $S = \{S_i\}^M$  标识子  $MLPRN$  对象集合, $S$  也是  $MLPRN$  类型的;当进程网没有下一层的子进程网时, $S=0$ ; $PRN = \{PRN_j\}^N$  为子进程网对象集合,当流程中没有子进程网对象时, $PRN=0$ ; $T$  标识要进行细化的进程网对象; $K$  标识进程网的层次数; $R$  为进行进程网细化时所遵守的细化规则。

层次结构的进程网的分层方法:将进程网的基本组件或者组合组件看作一个集合,原来的网系统放在上层,新生成的网系统或子网的细化留在下层,其他层的建立过程就是以上过程的递归,这样就形成了层次结构的进程网。

由于进程网有 5 种不同的组合运算:顺序、选择、循环、同步、异步<sup>[13]</sup>。在进行子网细化时,所建立的层次结构的进程网是形式多样的,通过这个多层次结构的进程网可以显示出系统的层次关系,有效地解决了当节点过多时的状态爆炸问题。

进程网的层次结构也可以理解成树形结构,顶层的业务流程在树的根节点,较为详细的业务流程在下层,即树的孩子节点。上层网模型是对整个系统的概括性描述和表示,下层模型是对某一部分进程的详细描述,是对上层模型的具体化。

#### 3.2 层次进程网细化的规则

(1)一个变迁扩展为一个子网的细化规则,如图 2 所示。

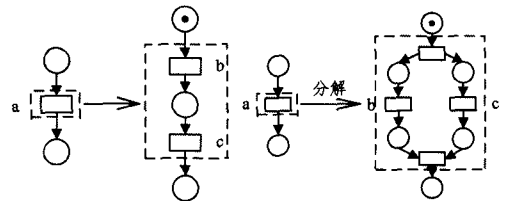


图 2 细化规则 1:一个变迁扩展为子网

(2)并发细化的细化规则,如图 3 所示。

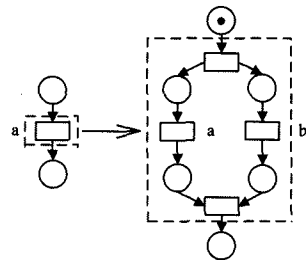


图 3 细化规则 2:进程网的并发细化规则

(3)选择细化的细化规则,如图 4 所示。

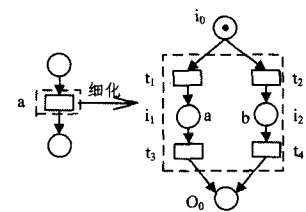


图 4 细化规则 3:进程网的选择细化规则

(4)循环细化的细化规则,如图 5 所示。

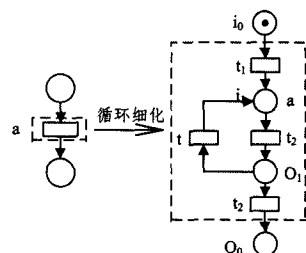


图 5 细化规则 4:进程网的循环细化规则

(5)通信细化的细化规则,如图 6 所示。

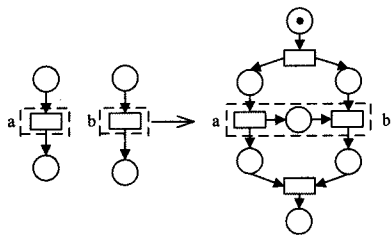


图6 细化规则5:进程网的通信细化规则

现有工作不支持进程网的分层建模。为了支持层次进程网的建模,需要实现进程网的上述细化规则,在现有工具中增加一般进程网的分层选中功能是一项核心任务。

## 4 层次进程网的建立过程

层次进程网的建立过程可以是自顶向下和自底向上的方法<sup>[14]</sup>。自顶向下的方法是先创建顶层的进程网粗略模型,再对其某一重要部分进行分析和细化。其实质是,用需要的细化规则替代顶层流程,得出细化后的系统层次进程网的模型。自底向上的建立过程是,先分析出重要的进程,用进程网模型建立底层的模型,然后再逐步将底层模型进行合并,并向上抽取形成顶层的流程。

层次进程网建模自顶向下的建立过程如图7所示。

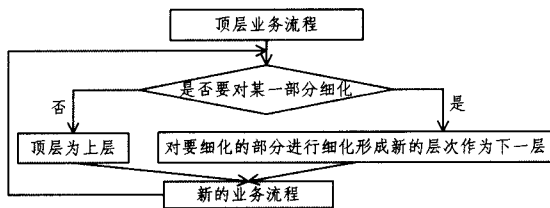


图7 层次进程网建模过程

从图7可以看出,其建模过程可以概括为:首先给出粗略模型,然后再对进程网的粗略模型进行细化,得到细化进程网模型。还可以对某些最底层(最基本)进程网细化,得到更为精细的进程网模型,即层次进程网模型。这是一种逐步建立可靠模型的方法。

传统的Petri网模型<sup>[15]</sup>没有提高系统的仿真能力<sup>[16]</sup>。通过以上建模过程可以发现,具有层次结构的进程网比Petri网更适合描述面向服务的系统,更适合表达服务组合的系统,同时也遵循了分而治之的原则,可以满足不同粒度的建模要求。

## 5 进程网分层的实现

### 5.1 二叉树在进程网分层中的应用

进程网建模过程是基本组件的两两组合,形成一种层次结构。为支持上文中的细化规则,需要选中当前进程网模型的基本元素。本文采用如下机制:在实现的过程中采用类似于哈夫曼树的构造过程<sup>[17]</sup>,先由底层往上构建,选择要进行组合运算的两部分,将这两部分分别作为左右子树,将进行组合运算之后新生成的部分作为父节点,然后再重复以上过程,直到建模完成为止。所以可以用二叉树的数据结构来模拟这个操作的过程。

#### (1) 二叉树的建立

首先创建一个BinaryNode的类,在该类中定义二叉树的存储结构,将节点中存储的数据定义为复杂组件类型。

在面板上画出基本组件并存入一个list。当进行进程网的组件组合时,分别有5种不同的组合方法:顺序、选择、完全并发、同步或异步并发、循环。选择需要的组合方式进行组合,将每次进行组合的两部分看作是二叉树的两个节点,新生成的组件是二叉树的根节点。将组合好的组件的根节点存入list中。

#### (2) list的更新

list中存储的是面板上当前的所有的根节点,当进行组合操作之后,将这两个选中的要进行组合的组件从list中删除。

#### (3) 坐标Position的更新

两个组件进行组合时,有5种不同的组合方式,组合之后两个组件的Position都发生了变化。需要分别针对这5种不同的组合方法计算出组合好之后的组件1、组件2的Position的值。

#### (4) 调用分层选中算法,实现进程网的分层功能。

### 5.2 进程网分层选中算法描述

#### 算法描述:

输入:鼠标在面板上单击后的坐标值以及当前选中的二叉树节点。参数列表如下:(x, y, currentRoot),其中x, y分别代表鼠标单击后的坐标值,currentRoot代表当前点中的节点

输出:当前选中的二叉树的节点

1) 函数getComponentNode(x, y, currentnode)

其功能是返回选中的节点。

1. getComponentNode(x, y, currentnode)

//x, y为double类型, currentnode为BinaryNode类型

2. if null == currnode

then return getComponentforestNode(x, y);

3. endif

4. childnode ← currentnode.getLeftNode();

5. if null != childnode

then if(childnode.getStartx <= x and childnode.getStartx <= y and childnode.getEndx <= x and childnode.getEndx <= y)

then return childnode;

6. endif

7. endif

8. childnode ← currentnode.getRightNode()

9. if null != childnode;

then if(childnode.getStartx)

then return childnode;

10. endif

11. endif

12. return getComponentNode(x, y);

2) 函数getComponentforestNode(x, y)

此函数可以实现按层次进行选择。

1. getComponentforestNode(x, y)

//x, y为double类型

2. if(componentforestNode.isEmpty)

3. then return null;

4. end if

5. for component.size() ← 0 to component.size()

6. if(componentforest.get(i).getStartx <= x and componentforest.get(i).getStartx <= y and componentforest.get(i).getEndx >= x and componentforest.get(i).getEndx >= y)

7. then

```
return getTreeClickNode(x,y,componentforest.get(i));
```

8. end if

9. return null;

在本算法中有一个 for 循环和递归调用,算法的时间复杂度为  $O(n^2)$ 。在实现按层次选中时,用到了栈的操作——出栈、进栈,使用二叉树的存储结构,空间复杂度为  $O(\log_2 n)$ 。

3) 函数 `getTreeClickNode(x,y,root)`

该函数用来得到选中的节点。

1. `getTreeClickNode(x,y,root)`

```
//x,y 为 double 类型,root 为 BinaryNode 类型
```

2. if null == root

3. then return null;

4. endif

5. if (root.getStartx <= x and root.getStary <= y and root.getEndx >= x and root.getEndy >= y)

```
then return root;
```

6. endif

7. left ← `getTreeClickNode(x,y,getLeftChildNode());`

8. if null != left

```
then return left;
```

9. endif

10. right ← `getTreeClickNode(x,y,getRightChildNode());`

11. return right;

12. return null;

4) 在 `DrawMouseListener` 类中的鼠标单击事件中调用函数 `GetComponentNode(x,y,z);`

```
call GetComponentNode(x,y,binarynodechoosingcomponent)
```

```
//x,y 为 double 类型,binarynodechoosingcomponent 为 BinaryNode 类型。
```

## 6 实验结果与分析

用进程网进行建模时,先自底向上选择两部分进程网进行运算,构建进程网的粗略模型。然后采用自顶向下的方法,利用前面建立起最顶层的进程网模型,再选取进程网的某部分进行细化。细化时就按照 3.2 节所提到的细化规则进行。图 8—图 12 示出了当用进程网进行顺序组合建模时的层次结构。

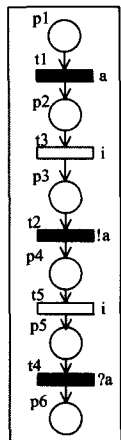


图 8 第 1 层

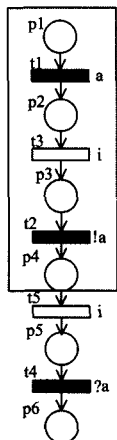


图 9 第 2 层

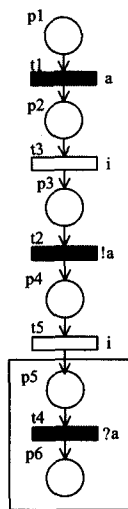


图 10 第 2 层

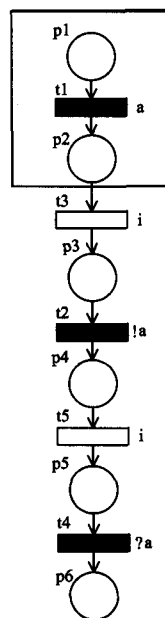


图 11 第 3 层

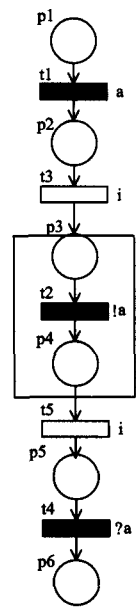


图 12 第 3 层

进程网进行顺序、选择组合仿真时 3 层层次结构如图 13—图 17 所示。

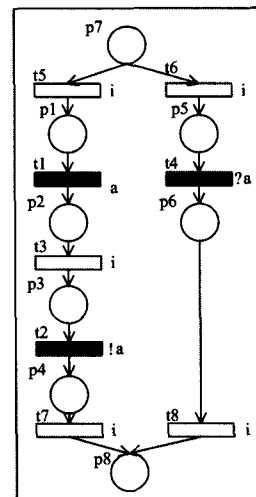


图 13 第 1 层

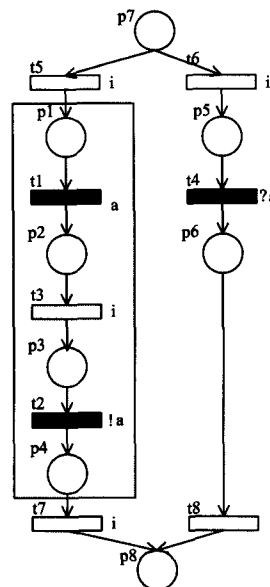


图 14 第 2 层

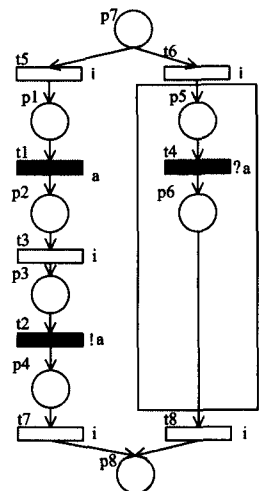


图 15 第 2 层

郭峰, 魏光, 邓蒙蒙. 一种面向服务 Petri 网模型及其结构化操作语义[J]. 小型微型计算机系统, 2013, 12(12): 2739-2740

[7] Fu Zuo-wei, Yue Xiao-bo, Wang Hui-ying. Workflow model based on extended weighted and timed Petri net[J]. Computer Engineering and Applications, 2014, 50(15): 55-58(in Chinese)

傅作为, 乐晓波, 王慧英. 加权的时间 Petri 网工作流模型研究[J]. 计算机工程与应用, 2014, 50(15): 55-58

[8] Hao Ke-gang. Open net-Model of interactive parallel system[J]. Journal of Northwest University (Natural Science Edition), 1997, 27(6): 461-466(in Chinese)

郝克刚. 开放网—交互式并行系统的模型[J]. 西北大学学报(自然科学版), 1997, 27(6): 461-466

[9] Hao Ke-gang. Object oriented model and software system theory [J]. Journal of Northwest University(Natural Science Edition), 1993, 23(5): 397-401(in Chinese)

郝克刚. 面向对象模型和软件系统论[J]. 西北大学学报(自然科学版), 1993, 23(5): 397-401

[10] Koutny M, Esparza J, Best E. Operational semantics for the Petri Box Calculus[M]// LNCS836: 5th International Conference Uppsala, Sweden; Springer Berlin Heidelberg, 1994: 219-220

[11] Wan He-ping, Wang Ming-zhe. Modeling and Analysis of Hierarchical Petri Net-based Workflow[J]. Computer Engineering and Applications, 2005, 41(15): 211-214(in Chinese)

万和平, 王明哲. 层次工作流 Petri 网建模与分析[J]. 计算机工程与应用, 2005, 41(15): 211-214

[12] Brauer W, Gold R, Vogler W. A Survey of Behavior and Equivalence Preserving Refinement of Petri Nets[C]// LNCS483: Advances in Petri Nets 1990. Berlin, Germany; Springer-Verlag, 1991: 1-46

[13] Zhou Yue-ming, Du Yu-yue, Liu Wei. Layer Modeling of Fault-tolerant System Based on Petri Nets[J]. Computer Engineering, 2010, 36(2): 39-41(in Chinese)

周月明, 杜玉越, 刘伟. 基于 Petri 网的容错系统分层建模[J]. 计算机工程, 2010, 36(2): 39-41

[14] Pu Fei, Lu Wei-ming. Preservation of Liveness and Deadlock-Freeness in Synchronous Synthesis of Petri Net Systems[J]. Journal of Software, 2003, 14(12): 1977-1978(in Chinese)

蒲飞, 陆维明. 同步合成 Petri 网系统活性与无死锁性的保持性[J]. 软件学报, 2003, 14(12): 1977-1978

[15] Liu Gang, Xia Chuan-liang. A Petri Net Refinement Method and its Applications[J]. Computer Science, 2006, 33(11A): 299-302(in Chinese)

刘刚, 夏传良. 一种 Petri 网精细化方法及其应用[J]. 计算机科学, 2006, 33(11A): 299-302

[16] Zhang Tao, Zhang Feng-lin, Tan Yue-jin. Multi-level PERT-Petri net model for process simulation[J]. Systems Engineering and Electronics, 2004, 26(1): 48-51(in Chinese)

张涛, 张凤林, 谭跃进. 流程仿真的多层次 PERT-Petri 网模型[J]. 系统工程与电子技术, 2004, 26(1): 48-51

[17] Fu Yong. An Algorithm in C++ for Building Huffman Tree with Min-Heap[J]. Computer Applications and Software, 2011, 28(3): 253-256(in Chinese)

付勇. 一个利用小顶堆构造哈夫曼树的 C++ 算法[J]. 计算机应用与软件, 2011, 28(3): 253-256

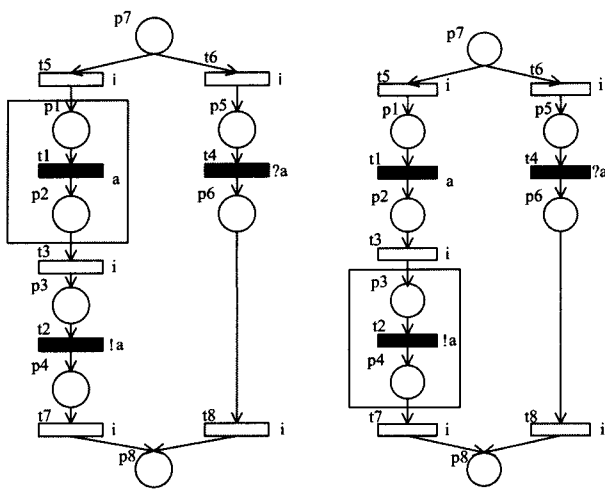


图 16 第 3 层

图 17 第 3 层

通过以上结果可以看出,不论是哪种组合运算之后得到的进程网模型,当要进行选中或者其他操作如细化时,第一次操作的都是第一层,当进行下一步操作时就进入到下一层,如此递归下去,直到操作到最底层时,再返回到顶层。这正好和进程网分层的思想一致,实现了进程网的分层操作。

由于篇幅有限,在进行其他组合运算,如完全并发、同步异步、循环之后的分层选中的示意图就不再一一给出。

**结束语** 本文提出了进程网分层的概念,并且采用了二叉树的存储结构实现了进程网的分层选中,研究了层次结构的进程网,为后续的细化操作打下基础。层次结构的进程网在进行系统业务流程仿真时有更强的描述能力。

目前实验室已经完成了进程网建模工具的画图、仿真、可达图的生成、分层选中。进程网的细化以及语义等价性等方面的研究还有待完善。

### 参考文献

[1] 吴哲辉. Petri 网导论[M]. 北京:机械工业出版社, 2006: 1-3

[2] Guo Feng, Deng Meng-meng, Yang Yan-gong. Design and implementation of process net modeling tool[J]. Journal of North China Univ. of Tech, 2014, 26(1): 1-2(in Chinese)

郭峰, 邓蒙蒙, 杨颜公. 进程网建模工具的设计与实现[J]. 北方工业大学学报, 2014, 26(1): 1-2

[3] Zhang Ji-jun, Wu Zhe-hui. Hierarchical Recursive Model of Petri Net[J]. Journal of System Simulation, 2003, 15 (Supplement): 90-92(in Chinese)

张继军, 吴哲辉. Petri 网的分层递归模型[J]. 系统仿真学报, 2003, 15(增刊): 90-92

[4] Ao Li-min, Wang Shu-da. A Developing Model for Software Workflow Based Hierarchical Petri Net[J]. Computer Science, 2004, 31(9A): 234-247(in Chinese)

敖丽敏, 王树大. 一种基于分层 Petri 网的软件工作流开发模型[J]. 计算机科学, 2004, 31(9A): 234-247

[5] Shen Bin, Yao Min, Yi Wen-sheng. Hierarchical Petri nets based software requirement refinement for Web service[J]. Journal of Zhejiang University (Engineering Science), 2006, 40(6): 1045-1050(in Chinese)

沈斌, 姚敏, 易文晟. 基于分层 Petri 网的面向服务需求求精[J]. 浙江大学学报, 2006, 40(6): 1045-1050

[6] Guo Feng, Wei Guang, Deng Meng-meng. A Service Oriented Petri Net Model and It's Structural Operational Semantics [J].