

А. В. Марков, А. А. Воевода

## ВЛОЖЕННЫЕ СЕТИ ПЕТРИ И НАГРУЖЕННЫЕ МЕТКИ: РАСПРЕДЕЛЕННАЯ СИСТЕМА И РЕКУРСИЯ<sup>1</sup>

Для моделирования параллельных процессов, сложных многоуровневых систем без дополнительных мест, переходов и их взаимосвязей, синхронизации разных потоков информации и представления в разных сетях одинаковых мест с разной маркировкой используются вложенные сети Петри. Альтернативой данного вида сетей являются сети с нагруженными метками, которые содержат дополнительную информацию о структуре системы или об истории её изменений. Сравнение подходов построения сетей Петри на основе вложенных сетей и сетей с нагруженными метками выполняется на примере анализа распределенной системы агентства проката автомобилей и решения рекурсивной задачи – нахождения факториала числа. Анализ распределенной системы агентства выполнен на основе двухуровневых вложенных сетей Петри с использованием вертикальной и горизонтальной синхронизации. Для сравнения осуществляется анализ на основе сетей с нагруженными метками в двух вариантах. В первом случае за основу берется структура системной сети, во втором – сеть состоит из трёх мест, к каждому из которых добавляется определенный набор типов данных для отображения возможных состояний системы. Нахождение факториала числа у вложенных сетей Петри осуществляется через вертикальную синхронизацию срабатывания переходов между системной сетью и элементными сетями, а при использовании нагруженных меток – через комбинированный тип данных мест сети. Переменная первого типа данных отображает значение величины искомого факториала, а во втором случае отслеживается значение текущей переменной, вычисленной на предыдущем шаге. Моделирование и автоматический анализ сетей Петри для данных примеров происходит в программной среде CPN Tools (Version 4.0.0).

**Ключевые слова:** вложенные сети Петри, нагруженные метки, рекурсия, распределенные системы, факториал, комбинированный тип данных, горизонтальная и вертикальная синхронизации срабатывания переходов.

### Введение

Для верификации проектируемых моделей на семантические ошибки прибегают к использованию сетей Петри [1], которые являются четверкой  $N = (P, T, F, m_i)$ , где  $P = \{p_1, p_2, \dots, p_n\}$  – множество мест;  $T = \{t_1, t_2, \dots, t_m\}$  – множество переходов, таких как  $P \cap T = \emptyset$ ;  $F \subseteq P \times T \cup T \times P$  – отношение;  $m_i: P \rightarrow N$  – начальная маркировка (размещение меток в местах сети Петри).

Анализ сетей Петри позволяет выявить тупиковые состояния  $m' \in V$ ,  $m' = m_n$ ,  $(m_n, t_n, m_{n+1})$ ,  $t_n \notin T$ , мертвые переходы<sup>2</sup>, заикливания  $m' \rightarrow *m'$ , а также проверять свойства ограниченности<sup>3</sup> сети, достижимости  $m \in [m_i]$  и покрываемости  $m'' \in [m']$  маркировок.

Одним из популярных способов анализа свойств сети Петри является построение и исследование графа состояний. Граф состояний представляется четверкой  $(V, E, src, trg)$ , где  $V$  – множество вершин;  $E$  – множество рёбер между вершинами;  $src, trg: E \rightarrow V$  – отображает в каждом ребре вершину, из которой оно получено, и вершину, к которой приводит выполнение срабатывания перехода, соответственно. Существует стандартный способ, когда происходит хранение информации обо всех посещенных состояниях сети, и альтернативный способ: метод плавающей линии [2], побитовое хеширование о состоянии [3], метод возможных ребер (comeback метод) [4], метод анализа пространства состояний [5–9].

<sup>1</sup> Работа выполнена при финансовой поддержке Минобрнауки России по государственному заданию № 2014/138. Тема проекта «Новые структуры, модели и алгоритмы для прорывных методов управления техническими системами на основе наукоемких результатов интеллектуальной деятельности».

<sup>2</sup> Переход называется мёртвым, если не существует достижимая от начальной разметки маркировка  $m' \notin [m_i]$ , при которой переход  $t \in T$  может сработать  $(m, t, m')$ .

<sup>3</sup> Сеть Петри  $(P, T, F, m_i)$   $k$ -ограничена, если и только если все маркировки  $m \in [m_i]$  и для всех  $p \in P: m(p) \leq k$ .

Существует большое количество классов (общие сети  $GN$ ; общие сети без петель  $LFN$ ; ординарные сети  $PN$ , бесконфликтные сети  $CFN$ , сети со свободным выбором  $FCN$ , маркированные графы  $MG$ , автоматные сети  $SM$ ) [10] и видов сетей Петри (временные сети, стохастические, ингибиторные, простые, цветные, иерархические, вложенные [11], возможные модификации и др.). Остановимся подробнее на вложенных сетях Петри, в частности, на двухуровневых системах, которые представлены набором:

$$NPN = (Atom, Lab, SN, (EN, \dots, EN), \Lambda),$$

где  $Atom = Var \cup Con$  – множество меток имен переменных  $Var = \{v, \dots\}$  и имен констант  $Con = Con_{atom} \cup Con_{net} = \{c, \dots\}$ ;  $Lab = Lab_v \cup Lab_h$  – множество меток, использующихся для вертикальной и горизонтальной синхронизации срабатывания переходов;  $(EN_1, \dots, EN_k) (k \geq 1)$  – конечный набор обыкновенных (элементных в  $NPN$ ) сетей Петри;  $\Lambda$  – частичная функция пометки переходов, выделяющая некоторые переходы системной сети метками  $Lab_v$  и некоторые переходы в элементных сетях метками  $Lab_v \cup Lab_h$ ;  $SN$  – сеть высокого уровня, которую представляют набором:

$$SN = (N, L, U, W, M_0),$$

где  $N = (P, T, F)$  – сеть;  $L = Expr(Atom)$  – язык выражений для представления переменных и констант;  $U$  – модель языка  $L$ ;  $W$  – функция, сопоставляющая каждой дуге  $(x, y) \in F$  некоторое выражение  $W(x, y) = (\theta_1, \dots, \theta_n)$  размерности  $n$ , где  $\theta_i \in L (1 \leq i \leq n)$  и  $n$  – арность позиции, инцидентной дуге  $(x, y)$ ;  $M_0$  – начальная разметка сети.

### Моделирование распределенной системы на основе вложенных сетей Петри

Одним из вариантов представления вложенных сетей Петри является использование списков в маркировке мест [12, 13], которой сходно со способами, предложенными в [14].

Отдельно выделим сети Петри с нагруженными метками<sup>1</sup> и [15], которые, по сути, являются альтернативным представлением вложенных сетей. В нагруженных метках информация о некоторых из позиций сети закладывается в тип данных некоторой метки. Данная информация интерпретирует количество меток в позиции, состояние или тип данных. Рассмотрим подробнее вложенные сети и сети с нагруженными метками на примере агентства по прокату автомобилей (рис. 1).

Вложенная сеть, моделирующая работу агентства, состоит из системной сети  $SN$  и двух элементных сетей  $EN_1$  и  $EN_2$ , представляющих клиентов и автомобили, соответственно. Для того, чтобы учитывать какой именно автомобиль получил конкретный клиент, удобно использовать в качестве фишек пары элементов. В нашем случае рассматриваются пары клиент–автомобиль. Соответственно сеть  $SN$  будет содержать фишки следующих типов: атомарные черные, сетевые – автомобили, сетевые – клиенты и пары маркированных сетей (клиент – автомобиль). В системной сети  $SN$  фишками в позициях  $s_1, s_4$  будут черные атомарные фишки (для позиций, представляющих пропозициональные условия); в позициях  $s_5, s_8$  – сетевые фишки<sup>2</sup> – автомобили; в позициях  $s_2, s_3$  – сетевые фишки – клиенты; в позициях  $s_6, s_7$  – пары сетевых фишек (клиент, автомобиль).

Приведем значения мест и переходов. *Места*:  $s_2$  – клиенты, желающие арендовать автомобиль;  $s_3$  – клиенты, обратившиеся за автомобилем в агентство;  $s_4$  – количество арендованных автомобилей;  $s_5$  – свободные автомобили;  $s_6$  – клиенты вместе с арендованными автомобилями;  $s_7$  – клиенты, которые хотят вернуть автомобиль в агентство;  $s_8$  – автомобили на мойке;  $c_1$  – клиенты, не нуждающиеся в автомобиле;  $c_2$  – клиенты, которые хотят арендовать автомобиль;  $c_3$  – клиенты, обратившиеся в агентство за автомобилем;  $c_4$  – клиенты, использующие автомобиль;  $c_5$  – клиенты, желающие вернуть автомобиль в агентство;  $c_6$  – клиенты, обратившиеся в агентство, чтобы вернуть автомобиль;  $a_1$  – автомобиль в гараже агентства;  $a_2$  – автомобиль

<sup>1</sup> Нагруженная метка – метка с дополнительной информацией о структуре системы или об истории ее передвижения.

<sup>2</sup> Сетевая фишка – фишка, перемещающаяся в системной и элементной сетях.

арендован;  $a_3$  – автомобиль используется;  $a_4$  – автомобиль находится на мойке. *Переходы:*  $l_1$  – клиенты, обратившиеся в агентство за автомобилем (агентство, клиент);  $l_2$  – отказ в аренде (агентство, клиент);  $l_3$  – передача автомобиля клиенту (агентство, клиент, автомобиль);  $l_4$  – обращение в агентство для возврата автомобиля (агентство, клиент, автомобиль);  $l_5$  – возвращение автомобиля в агентство (агентство, клиент, автомобиль);  $l_6$  – возвращение автомобиля в гараж агентства (агентство, автомобиль).

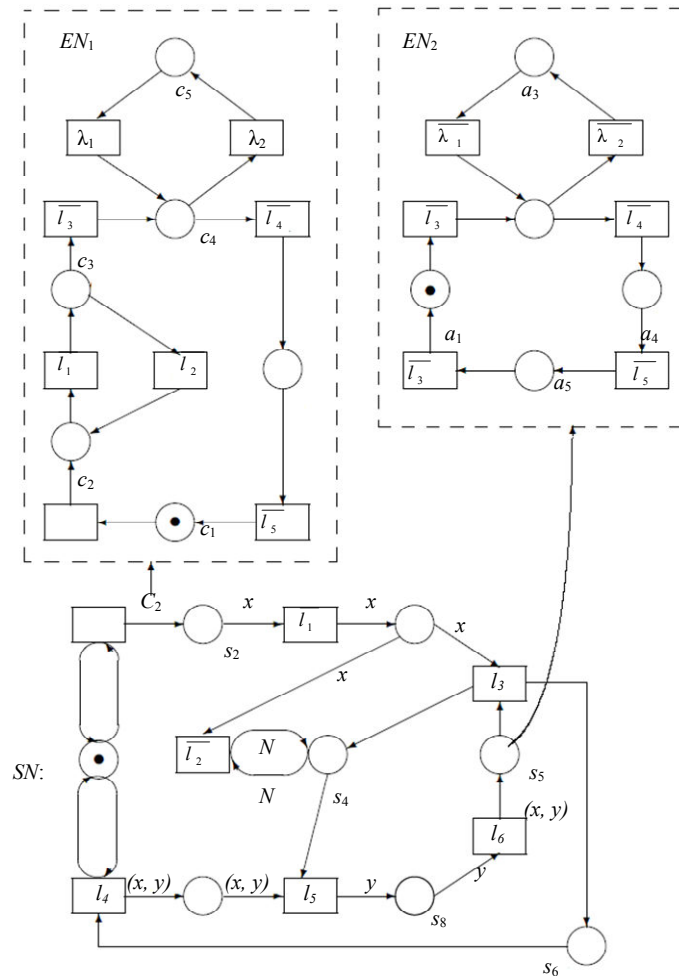


Рис. 1. Вложенная сеть Петри на примере агентства по прокату автомобилей (по [11])

В данном случае используются два вида синхронизации: вертикальная, когда переход в системной сети срабатывает одновременно с переходами в элементных сетях, и горизонтальная, когда одновременно срабатывают два перехода, находящихся в одной и той же позиции системной сети. Для их различия используют два вида меток:  $l, \bar{l}, \dots$  – для вертикальной синхронизации и  $\lambda, \bar{\lambda}, \dots$  – для горизонтальной.

Данная сеть Петри реализует модель, в которой клиент арендует автомобиль при его наличии в агентстве, использует и возвращает обратно. Таким образом, количество элементных сетей зависит от количества клиентов и автомобилей в агентстве.

#### Моделирование распределенной системы на основе сетей Петри с нагруженными метками

Реализацию данной сети при использовании нагруженных меток представим, используя два способа. В первом сохраним структуру системной сети, а информацию об изменении состояний в элементных сетях представим при использовании комбинированных типов данных (рис. 2).

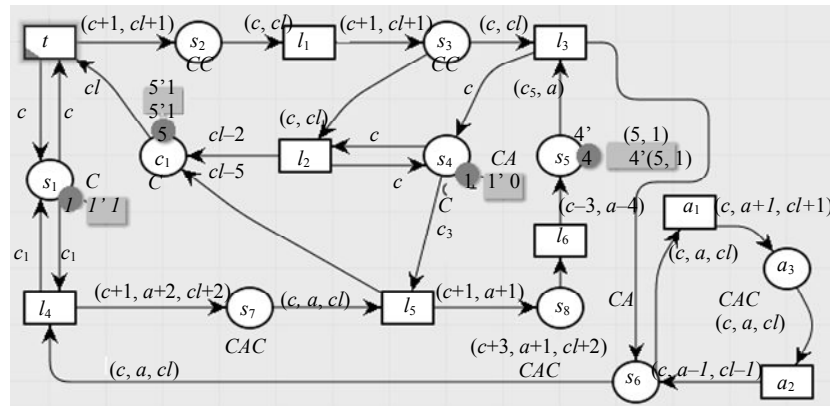


Рис. 2. Сеть Петри с нагруженными метками  
на примере агентства по прокату автомобилей, способ 1

При проектировании данной сети использовались следующие типы данных и переменные:

- $colset\ INT = int$ ;
- $colset\ CAC = product\ INT \times INT \times INT$ ;
- $colset\ CC = product\ INT \times INT$ ;
- $colset\ CA = product\ INT \times INT$ ;
- $colset\ C = INT$ ;
- $var\ cac : CAC$ ;
- $var\ cc : CC$ ;
- $var\ ca : CA$ .

Места элементарных сетей  $P_{EN_1} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$  и  $P_{EN_2} = \{a_1, a_2, a_3, a_4, a_5\}$  преобразуются в переменные, добавленные к маркировкам у соответствующих мест, т. к. элементарные сети интерпретируются при использовании нагруженных меток. Значение переменной соответствует индексу места, с которым она соотносится, например, в месте  $s_2$  находится фишка только со значением  $m_2 = (1(2, 2))$ . Для удобства состояния системной сети добавлены к типам данных, например, позиция  $s_7$  содержит фишки с комбинированным типом данных  $(INT \times INT \times INT)$ , т. е. можно отследить состояние клиента и автомобиля по отношению к агентству. Место  $a_3$  и переходы  $a_1, a_2$  добавляются к системной сети из элементарных  $\{s_6\} [a_1] \{a_3\}, \{a_3\} [a_2] \{s_6\}$ .

Изменение переменных происходит при срабатывании перехода за счет увеличения или уменьшения значений. Места  $c_1$  и  $s_5$  содержат пять и четыре фишки, соответственно, что означает пять потенциальных клиентов и четыре автомобиля в наличии у агентства.

Анализ сети проводился в программной среде CPN Tools Version 4.0.0, которая автоматически исследует пространство состояний с последующим сохранением результатов в текстовый файл (таблица).

#### Результаты анализа сети Петри

State Space. Nodes: 54, Arcs: 116, Secs: 0, Status: Full	Пространство имеет 54 состояния и 116 взаимосвязи между ними
Dead Markings – None	Тупиковых маркировок в сети нет
Dead Transition Instances – None	Мёртвых переходов в сети нет
Live Transition Instances – All	Все переходы живые, т. е. срабатывают при моделировании

Для реализации второго способа используется три места  $\{P = CAC\}$  (Company, Automobile, Client) и восемь переходов (рис. 3). Проектирование данной сети осуществляется на основе следующих типов данных и переменных:

- $colset\ INT = int$ ;
- $colset\ company = product\ INT \times INT \times INT \times INT \times INT \times INT \times INT \times INT$ ;
- $colset\ automobile = product\ INT \times INT \times INT \times INT \times INT$ ;
- $colset\ client = product\ INT \times INT \times INT \times INT \times INT \times INT$ ;

- $\text{var } c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8: \text{INT};$
- $\text{var } a_0, a_1, a_2, a_3, a_4, a_5: \text{INT};$
- $\text{var } cl_0, cl_1, cl_2, cl_3, cl_4, cl_5, cl_6: \text{INT}.$

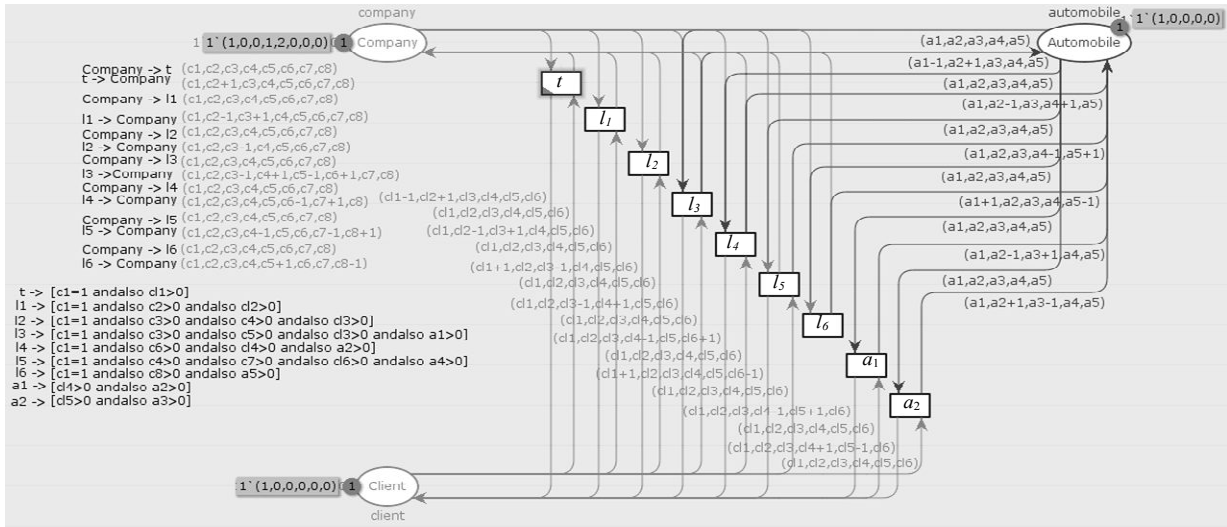


Рис. 3. Сеть Петри с нагруженными метками на примере агентства по прокату автомобилей, способ 2

В данной сети место Company имеет фишку, состоящую из восьми переменных, каждая из которых имеет свой тип данных (Integer). Значение переменной совпадает с количеством фишек в месте (индекс имени места совпадает с порядковым номером переменной). Переменные к соответствующим дугам места Company даны слева, как и условия на срабатывание переходов.

Необходимо отметить, что для реализации экземпляров у вложенных сетей необходимо построение новой сети, а для нагруженных меток – необходимо увеличить их количество в определенных местах или изменить значение переменной одного или нескольких типов данных. Условия срабатывания переходов, переменные и функции дуг сети становятся более сложными при использовании сетей Петри с нагруженными метками.

### Реализация рекурсии с использованием вложенных сетей Петри

В представленном ранее примере сеть имеет только один уровень вложенности. Но существует вложенность нескольких уровней [11]:

$$NPN = (\bar{N}, A_{atom}, I, N_0, M_0),$$

где  $\bar{N} = (N_0, \dots, N_k)$  – конечный набор сетевых компонент (сетевых фишек);  $A_{atom}$  – конечный набор сетевых фишек;  $I$  – интерпретация констант в выражениях на дугах в  $\bar{N}$ ;  $N_0$  – выделенная компонента (системная сеть);  $M_0$  – разметка системной сети  $N_0$  над  $\bar{N}$  и  $A_{atom}$ .

Данное свойство рассмотрим на примере реализации рекурсии [11–14], а именно – вычисления факториала (рис. 4).

Рекурсивная NP-сеть  $RN$ , изображенная на рис. 4, состоит из системной сети  $F_0$  и элементной сети  $F$ . Константа  $F$ , приписанная дуге, интерпретируется как элементная сеть  $F$  с начальной разметкой. Значения переменной  $x$  – сеть  $F$  с возможными (достижимыми) маркировками.

Исполнение рекурсивной  $RN$  происходит следующим образом. Изначально системная сеть имеет единственную фишку (черную точку) в позиции  $p_1$ . Первый шаг выбирается недетерминировано и зависит от начального значения аргумента  $n$  вычисляемой функции. Если  $n = 0$ , то срабатывает переход  $t_1$  (что соответствует вычислению  $fact(0) = 1$ ) и вычисление завершается.

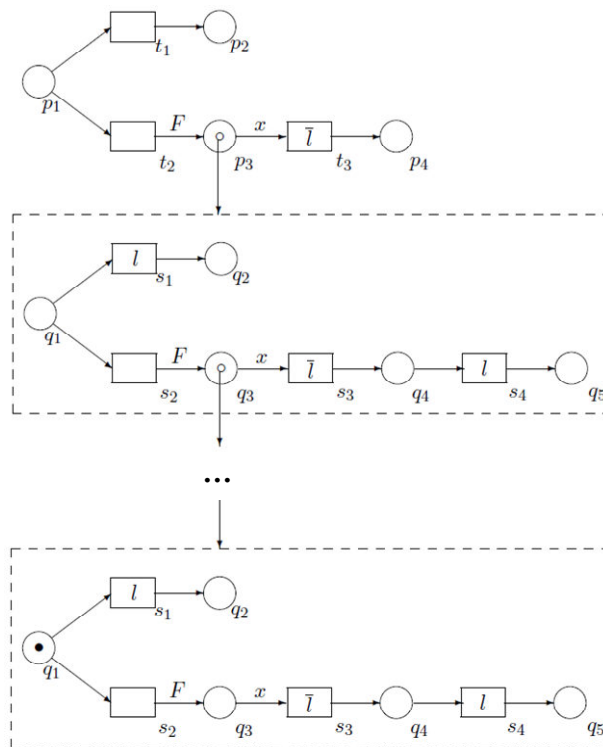


Рис. 4. Рекурсивная вложенная сеть Петри нахождения факториала числа (по [11])

Если  $n \neq 0$ , то срабатывает переход  $t_2$ , порождая элементную сеть  $F$  с начальной разметкой, что соответствует уменьшению  $n$  на 1 и внутреннему вызову процедуры вычисления факториала. После этого системная сеть  $F_0$  содержит сетевую фишку (сеть  $F$  с начальной разметкой) в позиции  $p_3$ . Далее в этой сетевой фишке может сработать переход  $s_2$ , что приведет к появлению новой сетевой фишки и нового уровня в сети  $RN$ . При срабатывании перехода  $s_1$  происходит синхронизация с переходом  $s_3$ , помеченным меткой  $\bar{l}$  в сети предыдущего уровня. Это приводит к исчезновению сетевой фишки самого нижнего уровня и уменьшению числа уровней в  $NP$ -сети на 1. При срабатывании перехода  $s_4$  в сетевой фишке (текущего) самого нижнего уровня происходит синхронизация с переходом  $s_3$  в «родительской» сети, что также уменьшит число уровней на 1. Этот процесс будет продолжаться до тех пор, пока в разметке не останется только один уровень – системная сеть с черной точкой в позиции  $p_3$ , после чего исполнение завершится.

#### Реализация рекурсии с использованием сетей Петри с нагруженными метками

При использовании нагруженных меток для вычисления факториала (рис. 5) используется комбинированный тип данных ( $INT \times INT$ ), переменная первого типа отображает значение числа искомого факториала, а второго – значение, вычисленное на предыдущем шаге.

В данном исполнении нахождение факториала происходит от искомого числа до единицы с уменьшением текущего значения на каждом шаге. Если значение на предыдущем шаге равно 0, то срабатывает переход Recursion 1, который декрементирует значение первой переменной и вычисляет вторую посредством произведения текущего значения числа, уменьшенного на 1. Если значение первой переменной не равно 1, а вторая переменная не равна 0, то сработает переход Recursion 2, который декрементирует значение первой переменной и вычисляет вторую посредством произведения текущего значения числа полученным на прошлом шаге произведением. Если первая переменная равна 1, то сработает переход Complete: в место Complete попадет фишка с найденным значением искомого факториала.

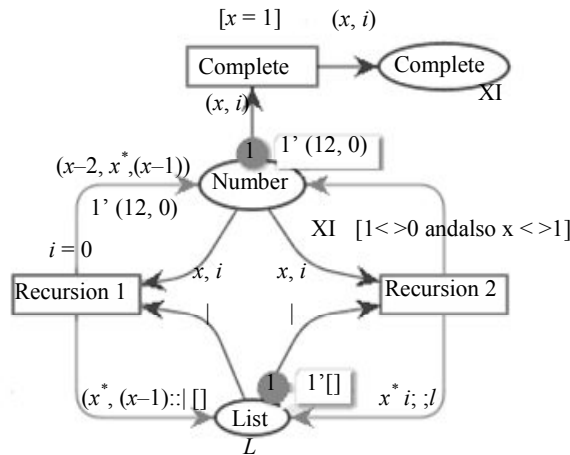


Рис. 5. Сеть Петри с нагруженными метками нахождения факториала числа

Из вышеизложенного следует, что лучше использовать нагруженные метки вместо вложенных сетей с вертикальной синхронизацией переходов.

### Заключение

На основе проведенных исследований вложенных сетей Петри и сетей с нагруженными метками сделаны предположения о подходящих областях их использования. Представлены положительные и отрицательные аспекты видов сетей на примере агентства по прокату автомобилей. Вложенные сети – это мощный инструмент, на основе которого показаны параллельные процессы, сложные многоуровневые системы без дополнительных мест, переходов и их взаимосвязей, синхронизация разных потоков информации и представление в разных сетях одинаковых мест с разной маркировкой, что весьма удобно. Отметим, что предпочтительным является применение сетей с нагруженными метками (метки с дополнительной информацией о структуре системы или об истории её изменений), если предполагается использование вложенных сетей одного или двух уровней.

Вложенные сети Петри выполняют рекурсивные задачи за счет вертикальной синхронизации срабатывания переходов, что показано на примере нахождения факториала числа. Отмечено, что при выполнении задачи более предпочтительным является использование сетей с нагруженными метками, чем вложенных сетей с вертикальной синхронизацией переходов.

### СПИСОК ЛИТЕРАТУРЫ

1. Романников Д. О. Разработка программного обеспечения с применением UML диаграмм и сетей Петри для систем управления локальным оборудованием: дис. ... канд. техн. наук / Д. О. Романников. Новосибирск, 2012. 195 с.
2. Christensen S. A. Sweep-Line Method for State Space Exploration / S. A. Christensen, L. M. Kristensen, T. Mailund // URL: <http://www.tzi.de/~edekamp/lectures/dmc/slides/sweep.pdf>.
3. Holzmann G. J. An Improved Protocol Reachability Analysis Technique / G. J. Holzmann // URL: <http://spinroot.com/spin/Doc/spe88.pdf>.
4. Westergaard M. Behavioural Verification and Visualisation of Formal Models of Concurrent Systems / M. Westergaard // URL: <https://westergaard.eu/wp-content/uploads/2009/10/thesis.pdf>.
5. Марков А. В. Анализ сетей Петри при помощи деревьев достижимости: / А. В. Марков, А. А. Воевода // Сб. науч. тр. Новосибир. гос. техн. ун-та. 2013. № 1 (71). С. 78–95.
6. Марков А. В. Анализ отдельных частей дерева достижимости сетей Петри / А. В. Марков // Сб. науч. тр. Новосибир. гос. техн. ун-та. 2013. № 3 (73). С. 58–74.
7. Воевода А. А. Инверсия простой ординарной сети Петри / А. А. Воевода, А. В. Марков // Науч. вест. Новосибир. гос. техн. ун-та. 2013. № 4 (53). С. 215–218.
8. Марков А. В. Инверсия сетей Петри / А. В. Марков // Сб. науч. тр. Новосибир. гос. техн. ун-та. 2013. № 4 (74). С. 97–121.
9. Марков А. В. Проверка достижимости маркировки сетей Петри при помощи инвертирования деревьев состояний для протокола передачи данных / А. В. Марков, А. А. Воевода // Докл. Томск. гос. ун-та систем управления и радиоэлектроники. 2014. № 1 (31). С. 143–148.

10. Бандман М. К. Территориально-производственные комплексы: прогнозирование процесса формирования с использованием сетей Петри / М. К. Бандман, О. Л. Бандман, Т. Н. Есикова. Новосибирск: Наука, 1990. 303 с.
11. Ломазова И. В. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой / И. В. Ломазова. Москва: Научный мир, 2004. 208 с.
12. Дворянский Л. В. Имитационное моделирование и верификация вложенных сетей Петри с использованием CPN Tools / Л. В. Дворянский, И. В. Ломазова // Моделирование и анализ информационных систем. 2012. Т. 19. № 5. С. 115–130.
13. Марков А. В. Моделирование процесса поиска пути в лабиринте при помощи сетей Петри / А. В. Марков // Сб. науч. тр. Новосибир. гос. техн. ун-та. 2010. № 4 (62). С. 133–140.
14. Марков А. В. Поиск манипулятором кратчайшего пути в лабиринте / А. В. Марков // Сб. науч. тр. Новосибир. гос. техн. ун-та. 2011. №4 (66). С. 75–90.
15. Воевода А. А. Рекурсия в сетях Петри / А. А. Воевода, А. В. Марков // Сб. науч. тр. Новосибир. гос. техн. ун-та. 2012. № 3 (69). С. 115–122.
16. Марков А. В. Понятие рекурсии в сетях Петри: факториал числа, числа Фибоначчи / А. В. Марков, А. А. Воевода // Сб. науч. тр. Новосибир. гос. техн. ун-та. 2013. №1 (71). С. 72–77.

Статья поступила в редакцию 02.02.2015,  
в окончательном варианте – 27.03.2015

### ИНФОРМАЦИЯ ОБ АВТОРАХ

**Марков Александр Владимирович** – Россия, 630073, Новосибирск; Новосибирский государственный технический университет; аспирант кафедры «Автоматика»; [muviton3@gmail.com](mailto:muviton3@gmail.com).

**Воевода Александр Александрович** – Россия, 630073, Новосибирск; Новосибирский государственный технический университет; д-р. техн. наук, профессор; профессор кафедры «Автоматика»; [ucit@ucit.ru](mailto:ucit@ucit.ru).



*A. V. Markov, A. A. Voevoda*

### NESTED PETRI NETS AND LOADED INSCRIPTION LABELS: DISTRIBUTED SYSTEM AND RECURSION

**Abstract.** For modeling concurrent processes, complex multi-level systems without additional places, transitions, and their relationships, synchronization of different flows of information and views on different nets of the same places with different markings, nested Petri nets are used. An alternative to this type of the nets are nets with loaded inscription labels containing additional information about the structure of the system or history of its changes. Comparison of the approaches to construction of Petri nets based on the nested nets and nets with loaded inscription labels is made by the example of the analysis of the distributed system of car rent agency and the recursive task – finding factorial. The analysis of the distributed system of the agency is made on the basis of two-level nested Petri nets using vertical and horizontal synchronization. In order to compare, the analysis based on the loaded inscriptional labels is carried out in two ways. In the first case, the structure of the system net is taken as a basis; in the second case the net consists of three parts, to each of which a certain set of data types is added to display the possible system states. Finding factorial of the number  $y$  in the nested Petri nets is made through vertical synchronization of transitions between the system net and the element nets, and while using the loaded inscription labels it is carried out via the combined data type of the net parts. The variable of the first data type displays the number of the desired factorial, and in the second case the value of the current variable calculated in the previous step is observed. Modeling and automatic analysis of Petri nets for these examples occur in the software environment CPN Tools (version 4.0.0).

**Key words:** nested Petri nets, loaded inscription labels, recursion, distributed systems, factorial, combined data type, horizontal and vertical synchronization of transitions.



## REFERENCES

1. Romannikov D. O. *Razrabotka programmnogo obespecheniia s primeneniem UML diagramm i setei Petri dlia sistem upravleniia lokal'nym oborudovaniem. Dis. kand. tekhn. nauk* [Development of the software using UML diagrams and Petri nets for the systems of control of local equipment. Dis. cand. tech. sci.]. Novosibirsk, 2012. 195 p.
2. Christensen S. A., Kristensen L. M., Mailund T. *Sweep-Line Method for State Space Exploration*. Available at: <http://www.tzi.de/~edekamp/lectures/dmc/slides/sweep.pdf>.
3. Holzman G. J. *An Improved Protocol Reachability Analysis Technique*. Available at: <http://spinroot.com/spin/Doc/spe88.pdf>.
4. Westergaard M. *Behavioural Verification and Visualisation of Formal Models of Concurrent Systems*. Available at: <https://westergaard.eu/wp-content/uploads/2009/10/thesis.pdf>.
5. Markov A. V., Voevoda A. A. Analiz setei Petri pri pomoshchi derev'ev dostizhimosti [Analysis of Petri nets by means of accessibility trees]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2013, no. 1 (71), pp. 78–95.
6. Markov A. V. Analiz otdel'nykh chastei dereva dostizhimosti setei Petri [Analysis of the separate parts of the accessibility tree of Petri nets]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2013, no. 3 (73), pp. 58–74.
7. Voevoda A. A., Markov A. V. Inversiiia prostoi ordinarnoi seti Petri [Inversion of simple ordinary Petri net]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2013, no. 4 (53), pp. 215–218.
8. Markov A. V. Inversiiia setei Petri [Inversion of Petri nets]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2013, no. 4 (74), pp. 97–121.
9. Markov A. V., Voevoda A. A. Proverka dostizhimosti markirovki setei Petri pri pomoshchi invertirovaniia derev'ev sostoianii dlia protokola peredachi dannykh [Testing the accessibility of labeling of Petri nets using the inverting of state trees for the protocol of data transmission]. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniia i radioelektroniki*, 2014, no. 1 (31), pp. 143–148.
10. Bandman M. K., Bandman O. L., Esikova T. N. *Territorial'no-proizvodstvennye komplekсы* [Territorial and industrial complexes]. Novosibirsk, Nauka Publ., 1990. 303 p.
11. Lomazova I. V. *Vlozhennye seti Petri: modelirovanie i analiz raspredelennykh sistem s ob'ektnoi strukturoi* [Nested Petri nets: modeling and analysis of the distributed systems with the objective structure]. Moscow, Nauchnyi mir Publ., 2004. 208 p.
12. Dvorianskii L. V., Lomazova I. V. Imitatsionnoe modelirovanie i verifikatsiia vlozhennykh setei Petri s ispol'zovaniem CPN Tools [Simulation and verification of the nested Petri nets using CPN Tools]. *Modelirovanie i analiz informatsionnykh sistem*. 2012, vol. 19, no. 5. pp. 115–130.
13. Markov A. V. Modelirovanie protsessa poiska puti v labirinte pri pomoshchi setei Petri [Modeling the process of searching the way in the labyrinth using Petri nets]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2010, no. 4 (62), pp. 133–140.
14. Markov A. V. Poisk manipulatorom krachaishego puti v labirinte [Search of the fastest way in the labyrinth using manipulator]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2011, no. 4 (66), pp. 75–90.
15. Voevoda A. A., Markov A. V. Rekursiia v setiakh Petri [Recursion in Petri nets]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2012, no. 3 (69), pp. 115–122.
16. Markov A. V., Voevoda A. A. Poniatie rekursii v setiakh Petri: faktorial chisla, chisla Fibonachchi [The notion of recursion in Petri nets: factorial, Fibonacci number]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta*, 2013, no. 1 (71), pp. 72–77.

The article submitted to the editors 02.02.2015,  
in the final version – 27.03.2015

## INFORMATION ABOUT THE AUTHORS

**Markov Alexander Vladimirovich** – Russia, 630073, Novosibirsk; Novosibirsk State Technical University; Postgraduate Student of the Department "Automation"; [muviton3@gmail.com](mailto:muviton3@gmail.com).

**Voevoda Alexander Aleksandrovich** – Russia, 630073, Novosibirsk; Novosibirsk State Technical University; Doctor of Technical Sciences, Professor; Professor of the Department "Automation"; [ucit@ucit.ru](mailto:ucit@ucit.ru).

