

ОБ ИСПОЛЬЗОВАНИИ ПРОГРАММНОГО ПАКЕТА CPN TOOLS ДЛЯ АНАЛИЗА СЕТЕЙ ПЕТРИ*

Д.О. РОМАННИКОВ, А.В. МАРКОВ

В данной статье описывается математический аппарат сетей Петри. Приведены правила построения, разновидности, а также свойства сетей. Продемонстрированы возможности программы CPN Tools, предназначенной для моделирования сетей Петри. Описаны механизмы построения и анализа полученных сетей.

Ключевые слова: инженерия ПО, сети Петри, CPN Tools, дерево достижимости, пространство состояний.

ВВЕДЕНИЕ

Для анализа систем на этапе проектирования пользуются математическим аппаратом сетей Петри, с помощью которого можно оценить корректность построения и своевременно отследить возможные ошибки, допущенные разработчиками. Для виртуальной имитации сетей Петри использовался программный пакет CPN Tools, который содержит необходимый функционал для моделирования и анализа созданных сетей.

1. СЕТЬ ПЕТРИ

В основе *сетей Петри* лежат два непересекающихся множества элементов: *места*, которые интерпретируются как пассивные элементы (состояния, ресурсы, накопители, каналы и т. д.) и *переходы*, которые интерпретируются как активные элементы (события, действия, выполнение оператора в программе, передача сообщения и т. д.). Места и переходы связаны между собой причинно-следственными связями. Условие наличия состояния, ресурса и т. д. в месте помечается с помощью метки (фишки, маркера) или меток в количестве равном емкости места. Графически места будут представлять как овалы, переходы как прямоугольники. Например, P1, P2, P3 – места, T1, T2 – переходы сети Петри, изображенной на рис. 1.

Дополнительно для каждого места и перехода имеется надпись, поясняющая назначение элемента сети (P1 – *initialState*, T1 – *on* и др.).

Срабатывание перехода, представляющего возникновение событие или выполнение действия, может зависеть от некоторых состояний и ресурсов. Наличие состояния или ресурса отображается с помощью метки. В начальной маркировке место P1 содержит одну метку. При срабатывании перехода T1 метка из P1 перейдет в место P2, при T2 – в P3.

Для дальнейшего изучения сетей Петри и их моделирования необходимо ознакомиться с правилами переходов. Для этого следует рассмотреть такое понятие сетей Петри, как кратность.

Кратностью входной/выходной позиции P для перехода T является число появлений позиции во входном/выходном комплексе перехода. Другими словами, кратность определяется числом дуг от перехода к месту и наоборот. Например, на рис. 2, а изображена сеть, в которой место P1 имеет выходную кратность 2 для перехода T1. Место P4 – входную кратность 2 для перехода T2 (см. рис. 2, б).

Сформулируем правила переходов:

1.

2) Переход может быть осуществлен, если все места, входящие в этот переход, имеют число меток не меньшее, чем их выходная кратность для этого перехода. Например, чтобы сработал переход T1, необходимо, чтобы место P1 содержало не меньше двух фишек (см. рис. 2, а).

2. Число фишек, которые появляются в месте при срабатывании перехода, равно входной кратности места. Если в месте P3 была одна фишка и сработал переход T2, то в месте P4 будет 2 фишки, а в P3 – ни одной (см. рис. 2, б).

При моделировании реальных систем классическая сеть Петри получается очень большой, что значительно затрудняет ее анализ. Дополнительной проблемой является невозможность моделирования многих аспектов объектно-ориентированных моделей служащих основой для реализации системы. Например, события и методы класса могут иметь параметры, для отображения которых нужны метки сложных (составных) типов. Для решения данной проблемы и многих других при моделировании программного обеспечения применяют различные расширения сетей Петри, но наибольшее распространение получили *раскрашенные* сети Петри.

Рассмотрим особенности раскрашенных сетей Петри. Для мест раскрашенных сетей Петри вводится такое понятие, как цвет (тип) места. Понятие цвет аналогично понятию типа в классических языках программирования высокого уровня и может быть простым: *bool*, *int*, *string* или составным (подобно классам и

структурам) – (*int, string*). Введение такого понятия значительно расширяет возможности сети Петри. Среди отличий раскрашенной сети от классической сети можно выделить следующее: местам и переходам назначается тип, показывающий тип метки, который может находиться в заданном месте; у меток появился новый параметр – значение меток, т. е. теперь важно не только количество меток, но и их значение; есть возможность накладывать на переход дополнительные ограничения. При переходе можно передавать не просто метку, а только требуемые параметры, входящие в состав этой метки.

2. СОЗДАНИЕ СЕТЕЙ ПЕТРИ В ПРОГРАММНОЙ СРЕДЕ CPN TOOLS

После запуска программного пакета *CPN Tools* появляется экран, разделенный на две части: слева меню, позволяющее выполнять работу с сетями, справа – рабочее поле, где располагается сеть и/или панель инструментов.

Для создания и работы с сетью необходимо выбрать инструменты: в левой части окна выбрать *Tool box* и перетащить (зажав левой кнопкой мыши) нужную панель инструментов в рабочую область.

На рис. 3 представлены основные панели, необходимые для моделирования в программной среде *CPN Tools*:

- *Net* – сетевые инструменты: для операций с сетями;
- *Create* – инструменты для создания элементов сети: для рисования и редактирования сетей Петри;
- *Simulate* – инструменты моделирования: для имитации поведения сети.

Создать новую сеть можно нажатием в левой области правой кнопки мыши и выбрать *New Net* (рис. 4) или воспользоваться панелью инструментов *Net* (рис. 3).

Пиктограммы на панели инструментов: *creates a new net* (создать новую сеть); *creates a new pages* (создать новую страницу); *closes a net* (закрыть сеть);

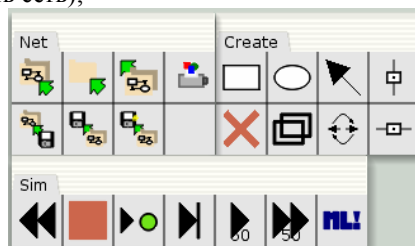


Рис. 3. Необходимые панели инструментов

prints a net colour, not showing current marking (печатать сеть); *loads in a net* (загрузить сеть); *saves a net* (сохранить сеть); *saves a net with a new name* (сохранить сеть как).

Для моделирования сети Петри понадобится панель инструментов *Create* (рис. 4). Панель *Create*: *creates a transition* (создать переход); *create a place* (создать место); *creates an arc*. (создать дугу); *creates a vertical magnetical guideline* (создать вертикальную опорную магнитную линию); *deletes an element* (удалить элемент); *clones an element* (клонировать элемент); *cycles between possible directions of arcs* (переключение возможных направлений дуги); *creates a horizontal magnetical guideline* (создать горизонтальную опорную магнитную линию).

Создать место или переход также можно при зажатии правой кнопкой мыши в окне *Binder* с последующим выбором *New Place* или *New Transition* соответственно (рис. 5, рис. 6).

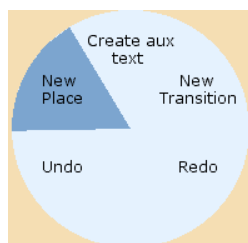


Рис. 5. Создание места

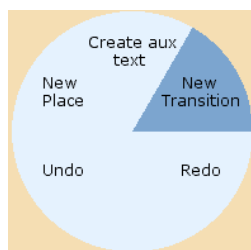


Рис. 6. Создание перехода

Для задания атрибутов места (имени, начальной маркировки и типа множества цветов) нужно выбрать место, назвать его, нажать клавишу *TAB*, снизу справа задается тип множества цветов, еще раз нажать клавишу

TAB, справа сверху задается начальная маркировка. Начальная маркировка задается следующим образом (рис. 7).

На рис. 7 видно, что место содержит 9 фишек: 1 фишку типа 1, 3 фишки типа 2, 5 фишек типа 3 (сложение мультимножеств: ++, сложение временных мультимножеств: +++), а «собака» показывает для каждого типа фишек, сколько этот тип задержался временных единиц, преодолевая переходы.

Переход имеет 4 типа атрибутов, такие как: имя перехода, условие запуска перехода, время задержки, сегмент кода. В данной работе использовалось 2 атрибута: имя перехода и время задержки (рис. 8). Атрибуты перехода задаются также как и атрибуты места (с помощью клавиши *TAB*).

Как видно, на рис. 8 имя перехода – T1, а время задержки равно 1.

Дуга создается с помощью панели инструментов *Create*.

Множество цветов выражения на дуге *expr* (рис. 9) должно соответствовать множеству цветов позиции, связанной с дугой. Если множество цветов выражения на дуге не соответствует множеству цветов позиции, связанной с дугой, то около дуги во время проверки синтаксиса появится сообщение об ошибке.

Для построения сети необходимо назначить переменные, которые будут «переносить» метки между местами, и типы переменных и мест. Каждая переменная должна принадлежать к какому-то определенному типу переменных. В *CPN Tools* есть встроенные типы переменных (*INT*, *STRING*). Для того чтобы создать дополнительный тип переменных, необходимо выбрать меню «*Declarations*» и добавить в нее строчку «*colset NAME = product TYPE*TYPE;*», где *NAME* – имя нового типа переменных, *product* – ключевое слово, определяющее составной тип данных, если тип не составной, то данное слово писать не нужно, *TYPE* – существующий тип переменных. Для создания переменной нужно в том же разделе меню вписать новую строчку «*var NAME1,NAME2: TYPE;*», где *NAME1*, *NAME2* – имена переменных, *TYPE* – тип переменных.

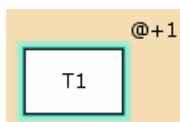


Рис. 8. Создание перехода

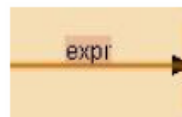


Рис. 9. Атрибут дуги

После создания переменных и указания типов их можно назначить в сети.

На рис. 10 цифрами обозначено следующее:

1. Описание метки в месте (1`1), первая цифра показывает количество меток, вторая – значение меток. Например, запись 20`10 показывала бы 20 меток со значением 10. Также в месте могут находиться несколько меток – 1`1++1`2 – означает, что в месте находятся 2 метки со значениями 1 и 2.
2. Тип места.
3. Переменная, которая «переносит» метку из одного места в другое. В данном примере из «*initaState*» в «*stopState*».

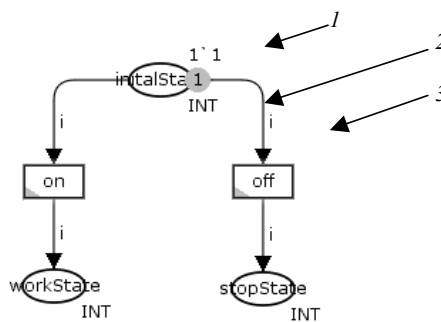


Рис. 10. Сеть Петри, в программном пакете *CPN Tools*

3. СВОЙСТВА СЕТЕЙ ПЕТРИ

Анализ сетей Петри осуществляется на основании их свойств. Основными свойствами, позволяющими судить о системе, являются:

1 *безопасность*. СП является безопасной, если при всех достижимых маркировках ее позиции не могут иметь более одной метки. Анализ безопасности проводится для исключения условий одновременного использования ресурса несколькими процессами;

2 *ограниченность*. Сеть называется ограниченной, когда все позиции являются ограниченными. В свою очередь, позиция является ограниченной, если количество меток в ней не может превышать целое число *K*.

Свойство ограниченности является важным, так как в реальных условиях число ресурсов ограничено, неограниченность места приводит к тому, что в нем будет происходить накопление меток до бесконечности, что является неприемлемым для модели реальной системы;

3 *живость*. Возможность срабатывания любого перехода при функционировании моделируемого объекта. Это свойство характеризует достижимость и функциональность всех переходов в сети. Т. е. за время работы все переходы срабатывают хотя бы один раз.

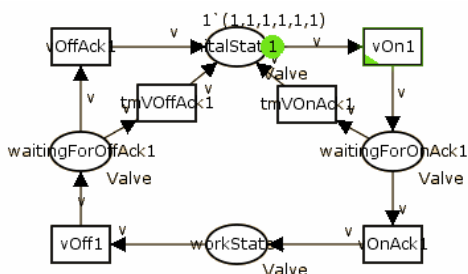


Рис. 11. Сеть Петри для анализа: часть рассматриваемой системы

Определить свойства сети можно на основании метода анализа основанного на анализе дерева достижимости.

Рассмотрим анализ свойств СП на примере тестовой сети изображенной на рис. 11. Дерево достижимости представляется множеством состояний сети, т. е. совокупностью всех возможных переходов из начальной маркировки. На рис. 12, а изображено дерево состояний при начальной маркировке M_0 . В начальной маркировке (1, 0, 0, 0) разрешен переход $vOn1$, который переведет систему в маркировку (0, 1, 0, 0). Из полученной маркировки можно снова попасть в начальную маркировку при помощи перехода $tmVOnAck1$. При переходе $vOnAck1$ получаем маркировку (0, 0, 1, 0). Второй шаг построения показан на рис. 12, б.

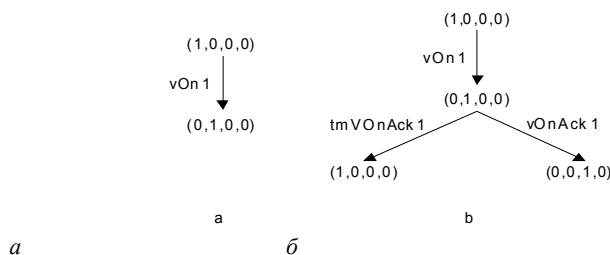


Рис. 12. Первый (а) и второй (б) шаги построения дерева достижимости

На рис. 13 показан третий шаг построения дерева достижимости, в котором при помощи переходов $tmVOffAck1$ и $vOffAck1$ переходят в начальную маркировку (0, 0, 1, 0).

Для построения деревьев состояний можно воспользоваться ресурсами программы *CPN Tools*. На рис. 14 показано дерево достижимости, полученное в программной среде *CPN Tools*.

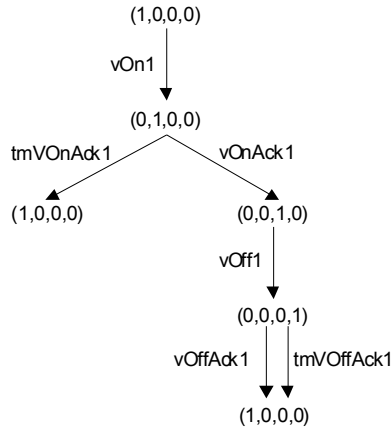


Рис. 13. Третий шаг построения дерева достижимости

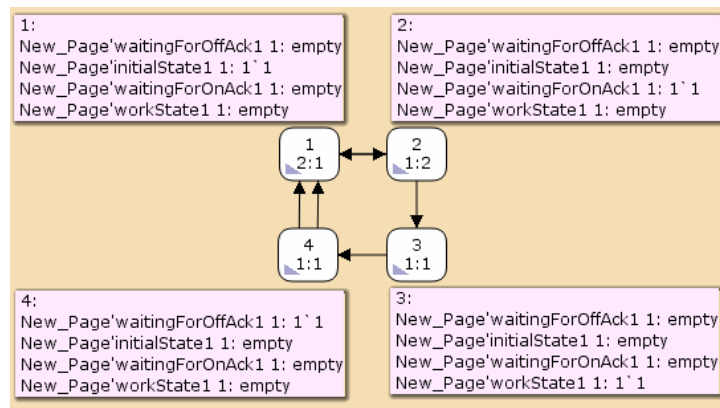


Рис. 14. Дерево достижимости, сгенерированное в среде CPN Tools

Повторение этой процедуры приводит к тому, что каждая достижимая маркировка окажется порожденной, а полученное дерево окажется бесконечным. Построение дерева состояний основывается на том принципе, что каждая последовательность переходов берет свое начало в корне дерева и оканчивается в какой-то предельно допустимой маркировке или может быть бесконечной. В тех случаях, когда нет возможности определить окончание ветки дерева, необходимо найти средства ограничения ветки до конечного размера.

Приведение к конечному представлению осуществляется двумя действиями:

1 использование пассивных маркировок, т. е. маркировок, которые не влияют на поведение логической части программы, а нужны для выявления и анализа свойств СП. Можно выделить следующие типы маркировок: терминальные вершины, дублирующие вершины (та маркировка, которая уже встречалась в анализируемом дереве).

2 рассматривается последовательность запусков переходов от начальной маркировки M_0 к одной из вариантов конечной маркировке $M' > M_0$. Маркировка M' связана с начальной маркировкой M_0 следующим соотношением: $M' = M_0 + (M' - M_0)$, $(M' - M_0) > 0$. Данные соотношения означают, что M_0 с расширена «дополнительными» метками в некоторых позициях. Последовательный запуск от начальной маркировки к конечной маркировке в результате даст маркировку $M_n = M_0 + n(M' - M_0)$.

Для сетей, где можно запустить один переход бесконечное количество раз подряд и получить в одном из мест бесконечное количество меток, которое обозначим символом « ω ». Для любого « a » определим:

$$\begin{aligned} M\omega + a &= \omega \\ \Pi \\ n\omega - a &= \omega \\ \Pi \\ 0a &< \omega \end{aligned}$$

Эти операции с символом « ω » достаточны для построения дерева.

4. АНАЛИЗ СЕТЕЙ ПЕТРИ

В ПРОГРАММНОЙ СРЕДЕ CPN TOOLS

Определение свойств сети Петри осуществляется при помощи программного пакета *CPN Tools*. Для анализа свойств сети Петри необходимо выбрать панель инструментов «SS», которая изображена на рис. 15.

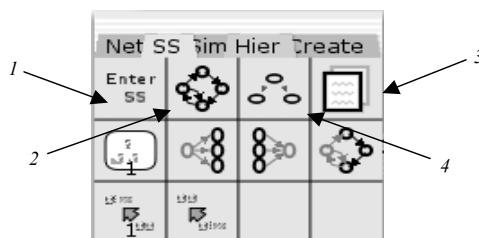


Рис. 15. Панель инструментов «SS» из программного пакета *CPN Tools*

Цифрами на рис. 15 обозначены:

- 1) инструмент, позволяющий создать пространство состояний для выбранной сети;
- 2) инструмент, позволяющий вычислить пространство состояний для выбранной сети;
- 3) инструмент, позволяющий вычислить граф состояний для выбранной сети;
- 4) инструмент, позволяющий получить отчет на основе вычисленных пространства состояний и графа состояний.

Для получения отчета о свойствах сети необходимо: создать пространство состояний (кнопка обозначена цифрой 1), вычислить пространство состояний (кнопка обозначена цифрой 2), вычислить граф состояний (кнопка обозначена цифрой 3), и сохранить отчет (кнопка обозначена цифрой 4).

Отчет о свойствах сети

Результаты отчета	Выводы
State Space Nodes: 4 Arcs: 6 Secs: 0 Status: Full	Пространство состояний модели вычислено полностью и содержит 4 узла, 6 дуг и 0 секции
Best Integer Bounds (Upper, Lower)	
VSub'initialStatel	1 0
VSub'waitingForOffAck1	1 0
VSub'waitingForOnAck1	1 0
VSub'workStatel	1 0
	Сеть ограниченная, так как все позиции сети являются ограниченными с границами равными 1

Окончание таблицы

Home Markings Initial Marking is a home marking marking Dead Markings None	Не все маркировки являются «домашними» – модель (алгоритм) не обратима. «Мертвых» маркировок в сети нет
Dead Transition Instances None Live Transition Instances All	В сети нет «мертвых» переходов

На основании отчета, представленного в таблице можно сделать выводы о свойствах сети.

ЗАКЛЮЧЕНИЕ

В предложенной работе детально рассмотрен математический аппарат сетей Петри. Описаны его основные свойства: безопасность, ограниченность, живость. Приводится пример анализа сети Петри через дерево достижимости и через анализ пространства состояний. Описаны основные элементы интерфейса и функционала программной среды *CPN Tools*, а также изложены основные принципы ее работы. Предложенные определения, описание свойств, методические указания работы в *CPN Tools* могут помочь при работе людям, столкнувшимся

с сетями Петри и *CPN Tools* впервые.

[1] *Зайцев Д.А., Шмелева Т.Р.* Моделирование телекоммуникационных систем в *CPN Tools*. – Одесса: Онат, 2006. – 60 с.

[2] *Питерсон Дж.* Теория сетей Петри и моделирование систем: Пер. с англ. – М.: Мир, 1984. – 264 с.

[3] *Котов В.Е.* Сети Петри. – М.: Наука, 1984. – 160 с.

Марков Александр Владимирович – аспирант кафедры автоматики Новосибирского государственного технического университета по специальности 05.13.01 «Системный анализ, управление и обработка информации (в промышленности)». Имеет 6 публикаций. E-mail: muviton3@mail.ru.

Романников Дмитрий Олегович – аспирант кафедры автоматики Новосибирского государственного технического университета по специальности 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей». Имеет 12 публикаций. E-mail: rom2006@gmail.com.

D.O. Romannikov, A.V. Markov

The review of works describing sharing UML diagrams and Petri nets at a design stage of software

In this article the mathematical apparatus of Petri nets is described. Rules of construction, a version, and also property of networks are given. Possibilities of the *CPN Tools* program intended for modeling of Petri nets are shown. Mechanisms of construction and the analysis of the received networks are described.

Key words: software engineering, Petri nets, *CPN Tools*, approachability tree, space of conditions.