

МОДЕЛИРОВАНИЕ ПРОЦЕССА ПОИСКА ПУТИ В ЛАБИРИНТЕ ПРИ ПОМОЩИ СЕТЕЙ ПЕТРИ*

А.В. МАРКОВ

Рассматриваются свойства параллельности в сетях Петри на примере нахождения выхода из лабиринта. По словесному описанию поставленной задачи была построена сеть Петри, предлагающая эвристический алгоритм поиска выхода из лабиринта. Явными достоинствами разработанной сети является то, что при изменении структуры лабиринта, структура сети остается прежней. Свойство параллельности сетей Петри наглядно было продемонстрировано при размножении метки на «перекрестках» и её движении в разных направлениях лабиринта.

Ключевые слова: инженерия ПО, *UML*-диаграммы, сети Петри, CPN Tools, пространство состояний, параллельные процессы, гиперпоточность, алгоритм древесного корня.

ВВЕДЕНИЕ

Проектирование программного обеспечения подразумевает выработку свойств системы на основе анализа постановки задачи, а именно: моделей предметной области, требований к ПО.

В зависимости от класса создаваемого ПО процесс проектирования может обеспечиваться как «ручным» проектированием, так и различными средствами его автоматизации. В процессе проектирования ПО для описания его характеристик используются различные нотации – блок-схемы, *ER*-диаграммы, *UML*-диаграммы, *DFD*-диаграммы, а также сети Петри. Данный вид проектирования позволяет переходить от словесного описания поставленной задачи к использованию вышеприведенных нотаций, а именно: сетям Петри и *UML*-диаграмм.

Использование *UML*-диаграмм совместно с сетями Петри [2, 4] дает возможность разработчикам выявить возможные неточности еще при постановки задачи. Анализ динамики функционирования системы можно описать через свободный язык сетей Петри [1].

Сети Петри – инструмент для исследования систем. Теория сетей Петри дает возможность моделирования системы математическим представлением ее в виде сети Петри. Предполагается, что при анализе сетей можно получить важную информацию о структуре и динамическом поведении моделируемой системы.

Несмотря на разнообразие моделируемых систем, выделяется несколько общих черт, которые должны быть отражены в особенностях используемой модели этих систем. Основная идея заключается в том, что системы состоят из отдельных взаимодействующих компонент. Каждая компонента сама может быть системой, но ее поведение можно описать независимо от других компонент системы, за исключением точно определенных взаимодействий с другими компонентами.

Действиям компонент системы присущи совмещенность или параллелизм. Действия одной компоненты системы могут производиться одновременно с действиями других компонент.

Совмещенная природа действий в системе создает некоторые трудности при моделировании. Поскольку компоненты системы взаимодействуют, необходимо установление синхронизации. Пересылка информации или материалов от одной компоненты к другой требует, чтобы действия включенных в обмен компонент были во время взаимодействия синхронизированы. Это может привести к тому, что одна компонента будет ждать другую компоненту. Согласование во времени действий различных компонент может быть очень сложным, а получающиеся в результате взаимодействия между компонентами трудны в описании.

Сети Петри разрабатывались специально для моделирования тех систем, которые содержат взаимодействующие параллельные компоненты, таким образом алгоритмы параллельного программирования и гиперпоточности можно протестировать с помощью сетей Петри [5].

1. ОПИСАНИЕ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ

Разрабатываемая система представляет собой лабиринт, по которому будет передвигаться метка. Лабиринт содержит проходы, стены, клетку входа и клетку выхода. Метка будет начинать движение с клетки «Начало» и заканчивать в клетке «Конец». За один шаг метка передвигается на одну клетку.

Лабиринт (рис. 1) задан двумерным массивом $x[i][j]$. Каждый элемент массива будет равен 1 или 0 (0 – проход, 1 – стена).

Создаем класс «Робот», который обращается к массиву «Лабиринт». Из него получаем информацию о своих координатах и возможных передвижениях. История передвижений будет сохраняться в метке.

При движении в самой метке сохраняется вся история передвижений, по которой можно отследить предыдущие перемещения. Также в метке должны указываться координаты местоположения.

На «Перекрестке» метка делится на возможное количество передвижений. Получившиеся метки будут двигаться в своем направлении одновременно, асинхронно.

Движение метки на предыдущее местоположение запрещено (в истории это последнее передвижение).

При попадании метки в тупик (нет возможных движений) свободные метки, если они существуют, продолжают движение, если отсутствуют, то выход из лабиринта не найден или не существует.

При попадании на клетку, уже пройденную другой меткой, первая метка будет считать это тупиком (после обращения к элементу массива равному 0, т. е. проход, данный элемент станет равным 2 – пройденное поле).

2. РЕАЛИЗАЦИЯ АЛГОРИТМА ПОИСКА ВЫХОДА ИЗ ЛАБИРИНТА С ПОМОЩЬЮ СЕТЕЙ ПЕТРИ

Смысл реализации этого алгоритма в том, что фишка «Robot» сразу размножается на 4 части, и в каждой из составных частей меняется координата (рис. 2). Переход «Comprasion n» работает в том случае, если координата в структуре массива свободна (присутствует фишка и она имеет символьный элемент равный «0»).

На рис. 3 представлена часть сети Петри на стадии выхода из лабиринта. Переход «Final position» работает в том случае, если место «Movement of Robot» будет содержать метку с координатой выхода из лабиринта. Если место «Movement of Robot» содержит метку с другими координатами, то работает переход «R2».

Лабиринт создан как двухмерный массив в месте «Structure of the labyrinth» с помощью составного множества цветов. Фишке «Robot» задаются начальные координаты, другими словами клетка входа. После срабатывания перехода «Comprasion n» фишка возвращается в место «Robot» и в место «Structure of the labyrinth» с измененным символьным типом на «2», для того чтобы другие метки не пересекали пути, пройденные другими метками.

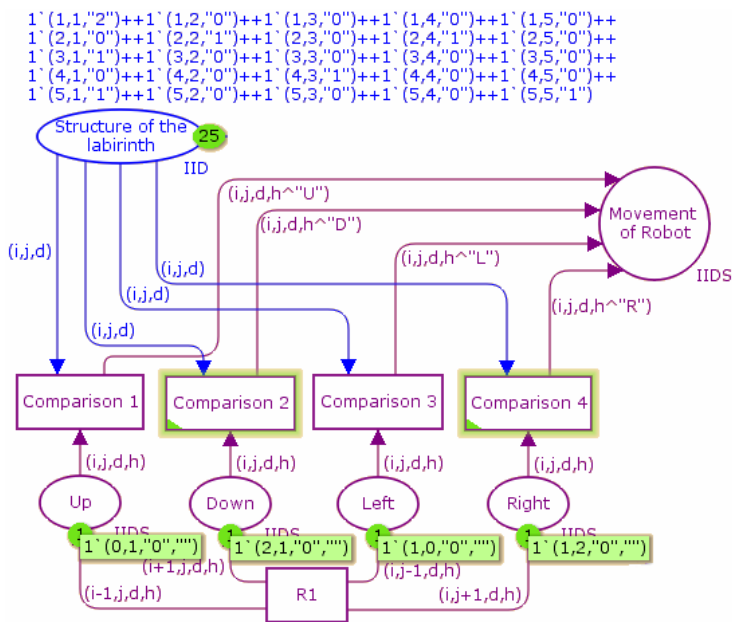


Рис. 2. Фрагмент сети Петри логика передвижения по лабиринту

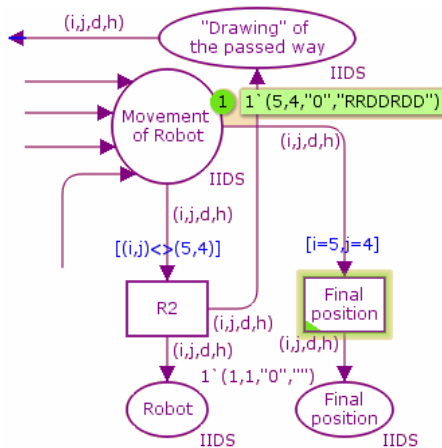


Рис. 3. Часть сети Петри на стадии выхода из лабиринта

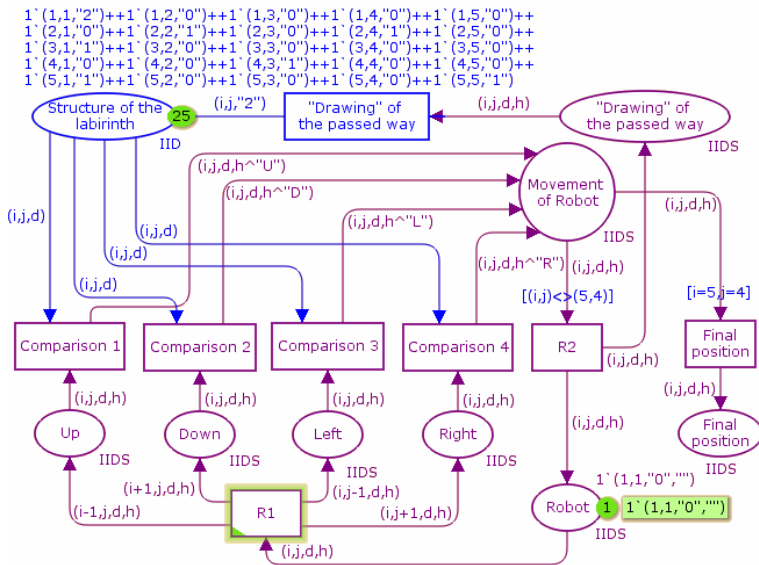


Рис. 4. Сеть Петри поиска выхода из лабиринта

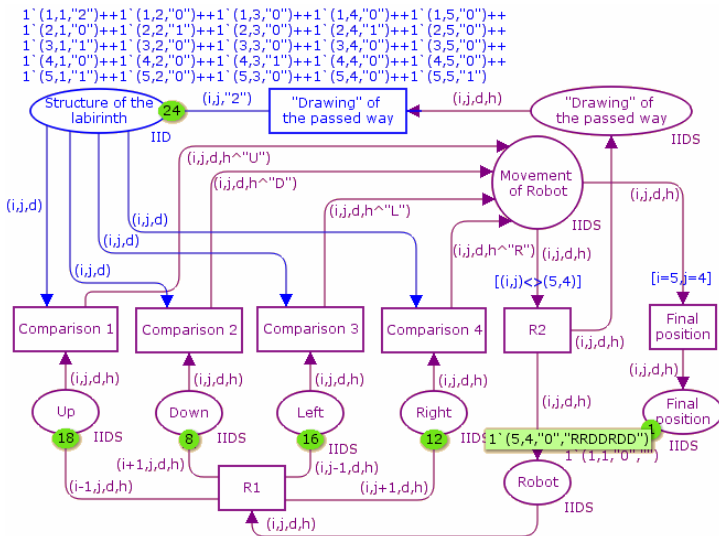


Рис. 5. Сеть Петри с найденным выходом из лабиринта

Переход «Final position» сработает только в том случае, если к нему обратится фишка с координатами поля выхода.

Предложенный эвристический алгоритм поиска выхода из лабиринта будет называться «алгоритмом древесного корня».

На рис. 5 представлена сеть с найденным выходом из лабиринта. В позиции «Final Position» находится метка, хранящая в себе координату местоположения и историю передвижений до клетки «Выхода». История накапливается при срабатывания перехода «Comparison n». «Structure of the labyrinth» показывает структуру лабиринта: «1» – стена, «0» – проход. По истории передвижений можно проследить путь движения метки до клетки «Выхода» из лабиринта.

3. АНАЛИЗ СВОЙСТ СЕТИ ПЕТРИ ПОИСКА ВЫХОДА ИЗ ЛАБИРИНТА

Реализация алгоритма была произведена в пакете CPN Tools, который позволяет проанализировать пространство состояний модели. Анализ пространства состояний полученной сети осуществлялся через автоматическую генерацию отчета.

Частичный стандартный отчет по пространству состояний модели, приведенной на рис. 4, имеет следующий вид:

CPN Tools state space report for:

<unsaved net>

Report generated: Mon Dec 27 22:48:59 2010

Statistics

State Space

Nodes: 16571

Arcs: 68950

Secs: 300

Status: Partial

Scc Graph

Nodes: 16571

Arcs: 68950

Secs: 3

Home Properties

Home Markings

None

Liveness Properties

Dead Markings

6188 [16571, 16570, 16569, 16568, 16567, ...]

Dead Transition Instances

None

Live Transition Instances

None

Fairness Properties

No infinite occurrence sequences.

Пространство состояний модели содержит 16571 узел, 68950 дуг, 300 секций, «мертвые» маркировки 6188 [6571, 16570, 16569, 16568, 16567, ...], в сети отсутствуют «мертвые» переходы.

ЗАКЛЮЧЕНИЕ

Предложенный алгоритм поиска выхода из лабиринта имеет ряд достоинств и недостатков. Основными положительными моментами являются следующие:

- 1) при изменении структуры лабиринта структура сети не измениться;
- 2) структура лабиринта может быть сколь угодно большой;
- 3) все свойства закладываются в метке.

К недостаткам можно отнести:

- 1) задание структуры лабиринта, массива очень громоздко;
- 2) возможно нахождение не самого «быстрого» выхода из лабиринта;
- 3) тупиковые метки остаются в сети.

Несмотря на существующие недостатки в сети, метка находит выход из лабиринта. Целью работы было с помощью данного примера показать свойство параллелизма в сетях Петри. При делении метки на «перекрестках» в лабиринте удалось наглядно продемонстрировать свойство параллелизма сетей Петри. Также удалось показать возможность получения сетей Петри по словесному описанию задания.

Как видно, представленная сеть нуждается в доработках. Последующими совершенствованиями сети станут:

- 1) задание структуры массива в файле Excel и импортирование его в сеть, так как задание массива в программной среде CPN Tools более трудоёмко;
- 2) ограничение количества последовательных движений метки до одного, т.е. при движении метки в следующем такте должна двигаться другая;

3) удаление тупиковых меток, возвращая при этом такую структуру лабиринта, которая была до внесения изменений тупиковой метки.

Автор выражает благодарность за постановку задачи и консультации при выполнении данного исследования профессору кафедры автоматики НГТУ Воеводе А.А.

[1] Воевода А. А. Марков А. В. О компактном представлении языков сетей Петри: сети с условиями и временные сети // Сб. науч. тр. НГТУ. – 2010. – № 2. – С. 77–83.

[2] Воевода А. А. Марков А. В. О Тестирование UML-диаграмм с помощью аппарата сетей Петри на примере разработки ПО для игры «Змейка» // Сб. науч. тр. НГТУ. – 2010. – № 3. – С. 51–61.

[3] Воевода А. А. Романников Д. О. Особенности проектирования систем реального времени при помощи UML и сетей Петри // Сб. науч. тр. НГТУ. – 2010. – № 3. – С. 51–61.

[4] Зимаев И.В. О возможности автоматической трансляции UML-диаграмм деятельности с сети Петри // Сб. науч. тр. НГТУ. – 2009. – № 1. – С. 57–63.

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

[6] Прытков Д. В. О применении сетей Петри для исполнения алгоритмов на примере решения задач о кратчайших путях с единственным источником // Сб. науч. тр. НГТУ. – 2010. – № 3. – С. 91–99.

[7] Шамим Э. Джейсон Р. Многоядерное программирование. – СПб.: Питер, 2010. – 316 с.

Марков Александр Владимирович – магистрант кафедры автоматики Новосибирского государственного технического университета по направлению инженерия ПО. E-mail: muviton3@mail.ru.

A.A. Markov

Modeling process of path searching in labyrinth using Petry net

Are considered property of parallelism in Petri nets on example of a finding of exit from a labyrinth. Under the verbal description of a task in view Petri nets offering heuristic algorithm of search of exit from a labyrinth has been constructed. Obvious advantages of the developed network is that at change of structure of a labyrinth, the network structure remains former. Property of parallelism in Petri nets has visually been shown at label reproduction at "cross-roads" and its movement divergently a labyrinth.

Key words: software engineering, *UML*-diagrams, Petri nets, CPN Tools, state space, parallel processes, hyper-threading, algorithm of an arboreal root.