СБОРНИК **НАУЧНЫХ ТРУДОВ НГТУ. – 2010. – № 3(65) –** 95–104

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 62-50:519.216

МОДЕЛИРОВАНИЕ ПРОЦЕССА ПОИСКА ПУТИ В ЛАБИРИНТЕ ПРИ ПОМОЩИ СЕТЕЙ ПЕТРИ ДЛЯ СИСТЕМЫ ИЗ ДВУХ СВЯЗНЫХ ЗВЕНЬЕВ*

А.В. МАРКОВ, А.А. ВОЕВОДА

Рассматривается свойство параллельности в сетях Петри на примере нахождения выхода из лабиринта. По словесному описанию поставленной задачи была построена сеть Петри для системы из двух связных звеньев. 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. Предложено дальнейшее направление развития данного алгоритма.

Ключевые слова: инженерия ПО, UML-диаграммы, сети Петри, CPN Tools, параллельные процессы, гиперпоточность, система из двух связных звеньев.

ВВЕДЕНИЕ

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

В зависимости от класса создаваемого ПО процесс проектирования может обеспечиваться как «ручным» проектированием, так и различными средствами его автоматизации. В процессе проектирования программного обеспечения 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 [3, 5–7, 9] дает возможность разработчикам выявить возможные неточности еще при постановке задачи. Анализ динамики функционирования системы можно описать через свободный язык сетей Петри [1,2].

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 компонентов. Каждый компонент сам может быть системой, но его поведение можно описать независимо от других компонентов системы, за исключением точно определенных взаимодействий с другими компонентами.

Действиям компонентов системы присущи совмещенность или параллелизм. Действия одного компонента системы могут производиться одновременно с действиями других компонентов.

Совмещенная природа действий в системе создает некоторые трудности при моделировании. Поскольку компоненты системы взаимодействуют, необходимо установление синхронизации. Пересылка информации или материалов от одного компонента к другому требует, чтобы действия включенных в обмен компонентов были во время взаимодействия синхронизированы. Это может привести к тому, что один компонент будет ждать другого компонента. Согласование во времени действий различных компонентов 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.

Опираясь на [3–9], разработана система, которая является модификацией системы, созданной ранее [4].

1. ОПИСАНИЕ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ

Разрабатываемая система представляет собой лабиринт (рис. 1), по которому будет передвигаться робот. Лабиринт содержит проходы, стены, клетку входа и клетку выхода. Роботом является конструкция длиной в одну клетку из двух звеньев, соединенных между собой. На рис. 2 представлено 8 существующих вариантов положения робота, конструкция может увеличиваться не больше чем на половину клетки.

Изначально робот занимает две клетки: первое звено находится в лабиринте, второе звено за его пределами. За один шаг каждое звено проходит по одной клетке.

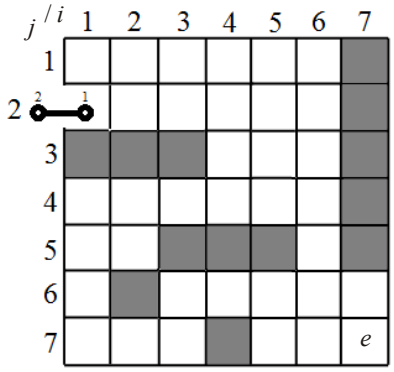


Рис. 1. Лабиринт

Лабиринт (рис. 1) задан двухмерным 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 (рис. 2).

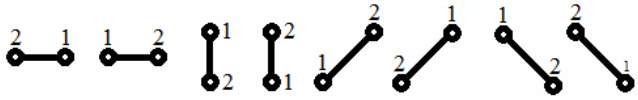


Рис. 2. Варианты положения робота

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

На «Перекрестке» робот делится на возможное количество передвижений, каждый из которых будет двигаться в своем направлении одновременно.

Движение робота на предыдущее местоположение запрещено (в истории движения – это последнее передвижение).

При попадании робота в тупик (нет возможных движений) свободные для перемещения роботы, 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 обращении к элементу массива, равному 0, т. е. проход, данный элемент станет равным 2, т. е. пройденное поле) [4].

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

Сеть Петри, представленная на рис. 3, включает в себя лабиринт, робот и возможные передвижения робота. Лабиринт включает в себя массив данных, реализуемый с помощью меток в месте “Structure of the labyrinth”, а также проверку завершения поиска выхода из лабиринта.

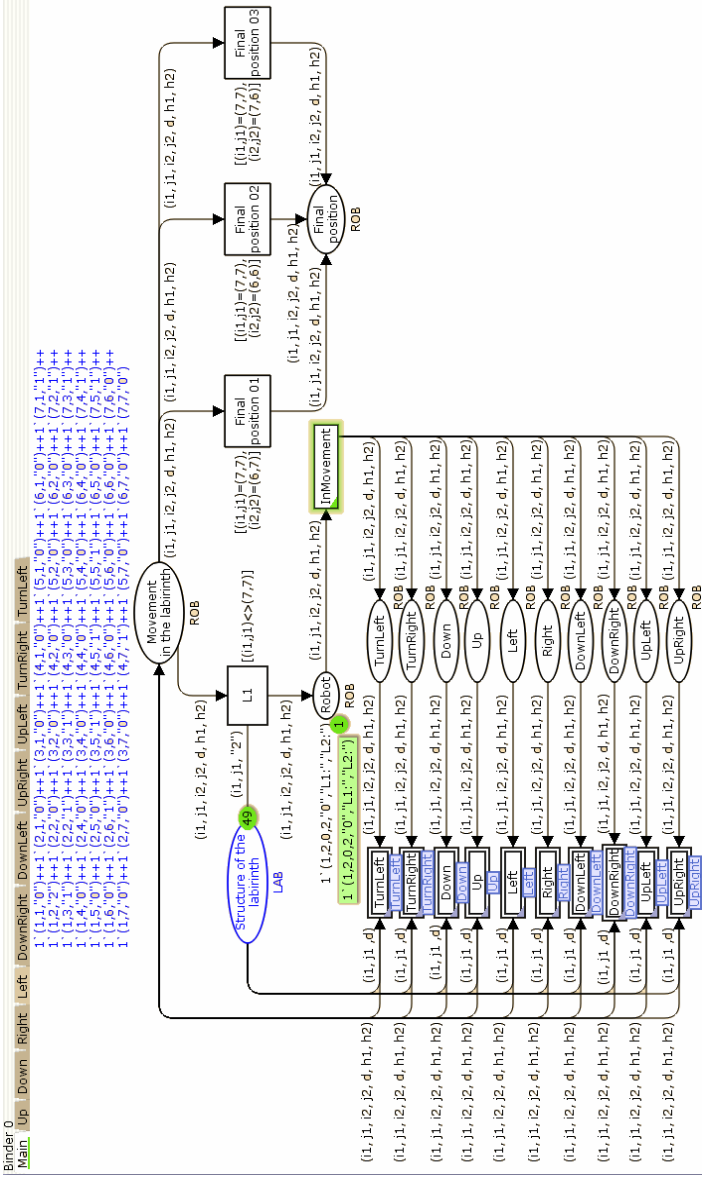
Смысл реализации данного алгоритма в том, что фишка «Robot» сразу размножается на 10 частей – это возможные передвижения и повороты. В свою очередь, каждое передвижение и поворот может менять координаты робота в зависимости от его положения в лабиринте. Поэтому для передвижений и поворотов созданы подстраницы, показывающие, как меняются координаты звеньев робота при передвижении или повороте в зависимости от положения самого робота. Пример подстраницы представлен на рис. 4.

Переход “Up 01 … 08”, “Down 01 … 08”, “Right 01 … 08”, “Left 01 … 08”, “DownRight 01 … 08”, “DownLeft 01 … 08”, “UpRight 01 … 08”, “UpLeft 01 … 08”, “TurnRight 01 … 08”, “TurnLeft 01 … 08” (любой переход движения подстраниц) сработает в том случае, если координата в структуре массива свободна (присутствует фишка и она имеет символьный элемент, равный “0”).

На рис. 5 представлена часть сети Петри на стадии начала поиска выхода из лабиринта. Как видно из рисунка, робот может иметь три положения, при которых будет найден выход из лабиринта. В переходах «Final position 01…03» заложены условия, соблюдение одного из них приведёт робота в финальное поле лабиринта.

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 «L1» фишка возвращается в место «Robot» и в место «Structure of the labyrinth» с измененным символьным типом на «2», для запрета движения по пройденному пути робота.

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



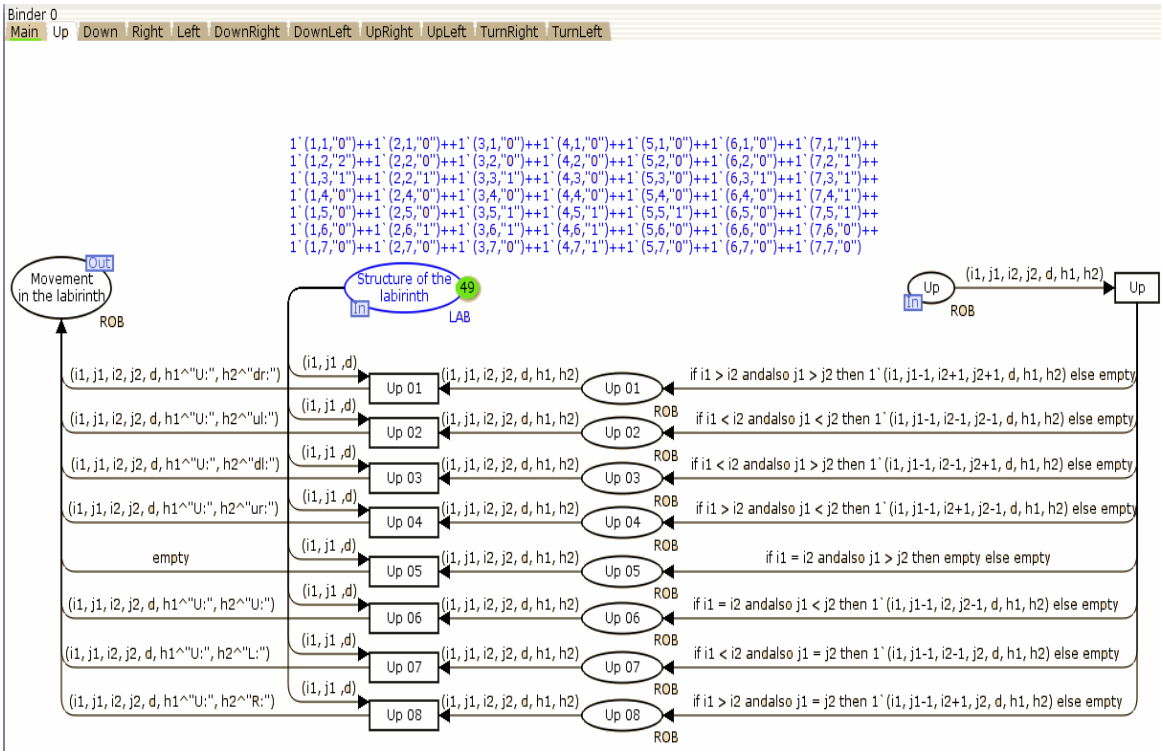


Рис. 4. Подстраница “Up” сети Петри нахождения выхода из лабиринта

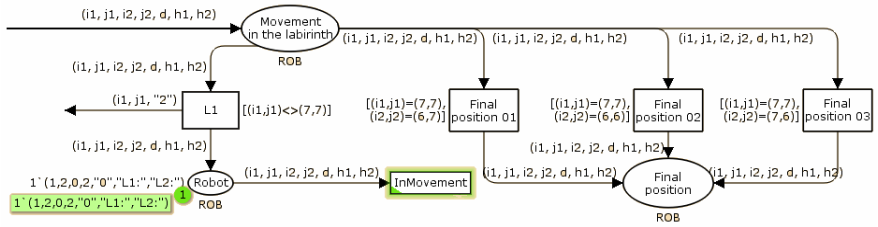
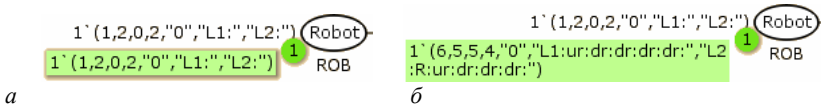


Рис. 5. Часть сети Петри на стадии выхода из лабиринта

Фишка робот имеет составное множество цветов, т.е. составное множество типов данных (рис. 6). Первые две цифры соответствуют координате первого элемента конструкции робота (*i*, *j*), третья и четвёртая цифры иллюстрируют положение второго звена в лабиринте. С помощью последних двух типов данных можно проследить историю передвижений первого и второго элементов соответственно.



а *б*

Рис. 6. Фишка-робот:

а – фишка-робот с начальными координатами, без истории передвижений; *б* – фишка-робот после нескольких шагов, очередность которых можно проследить по истории передвижения первого и второго элементов

На рис. 7 приведена часть главной страницы сети Петри с найденным выходом из лабиринта. В позиции «Final Position» находится метка, хранящая в себе координату местоположения первого и второго звена, а также историю их передвижений до клетки Выхода. По истории передвижений можно проследить путь движения метки до клетки «Выход» из лабиринта.

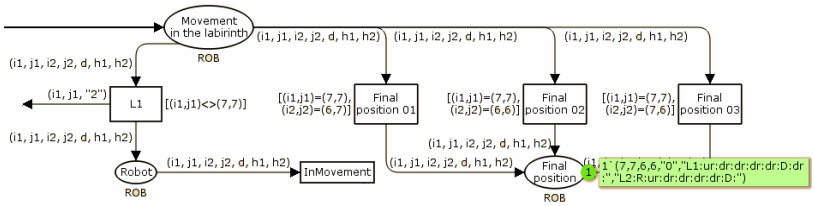


Рис. 7. Часть главной страницы сети Петри с найденным выходом из лабиринта

ЗАКЛЮЧЕНИЕ

Предложенный в данной работе алгоритм поиска выхода из лабиринта является модификаций алгоритма, представленного в работе [4].

Он унаследовал основные положительные моменты: при изменении структуры лабиринта структура сети не изменится, структура лабиринта может быть сколь угодно большой, все свойства закладываются в метке.

Тем не менее система претерпела значительные изменения, которые заключаются в модификации конструкции робота, а именно увеличение звеньев до двух и возникновении связи между ними длиной в клетку. Изменение структуры робота повлекло за собой добавление целочисленных и символьных типов данных для координирования звеньев и отображения истории их перемещений.

Также стоит отметить, что появление второго звена в конструкции робота привело к увеличению вариантов его передвижения и тем самым усложнило сеть.

Следующим шагом станет возможность увеличения длины конструкции до нескольких клеток, а также сохранения «свободного» места в лабиринте после прохождения его роботом.

При удлинении робота сузится выбор вариантов его передвижения по лабиринту, так как возникнут случаи, когда при свободном проходе, например вправо, робот повернуться в этом направлении не может из-за препятствия в виде «стенки» для перемещения соединительной конструкции между звеньями (рис. 8).

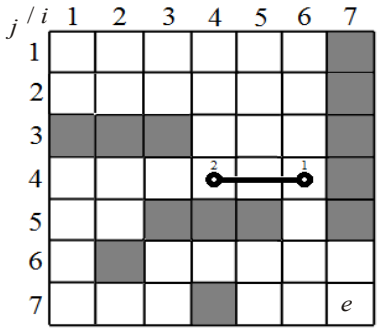


Рис. 8. Ситуация, при которой первый элемент
не может передвинуться на клетку ниже

Звено 1 не может переместиться на координату (6,5), из-за попадания соединительного элемента в координату «стенки» (5,5).

Сохранение «свободного» места после прохождения его роботом послужит возможностью проработать все варианты передвижения робота по лабиринту, что, в свою очередь, позволит найти наикратчайший выход из лабиринта.

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

[2] *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. 77–82.

[3] *Воевода А. А. Марков А. В.* Тестирование UML-диаграмм с помощью аппарата сетей Петри на примере разработки ПО для игры «Змейка» // Сб. науч. тр. НГТУ. – 2010. – № 3. – С. 51–60.

[4] *Марков А. В.* Моделирование процесса поиска пути в лабиринте при помощи сетей Петри // Сб. науч. тр. НГТУ. – 2010. – № 4. – С. 133–140.

[5] *Романников Д.О. Марков А. В. Зимаев И.В.* Обзор работ посвященным разработке ПО с использованием UML и сетей Петри // Сб. науч. тр. НГТУ. – 2011. – № 1. – С. 91–104.

[6] *Воевода А. А. Романников Д. О.* Особенности проектирования систем реального времения при помощи UML и сетей Петри // Сб. науч. тр. НГТУ. – 2009. – № 1. – С. 57–62.

[7] *Зимаев И. В.* О возможности автоматической трансляции UML-диаграмм деятельности в сети Петри // Сб. науч. тр. НГТУ. – 2010. – № 1. – С. 149–156.

[8] *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. 91–98.

[9] *Марков А. В.* Автоматизация разработки программного обеспечения с использованием сетей Петри: магистерская диссертация. – Новосибирск: НГТУ, 2011.

Марков Александр Владимирович – аспирант кафедры автоматики Новосибирского государственного технического университета. E-mail: muviton3@mail.ru.

Воевода Александр Александрович – профессор кафедры автоматики Новосибирского государственного технического университета. E-mail: voevoda@ucit.ru.

A.V. Markov, A.A. Voevoda

Modeling process of path searching in labyrinth using Petri nets for system from two connected links

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 net for system from two connected links. Obvious advantages of the developed net 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 "crossroads" and its movement divergently a labyrinth. Further direction of development of this algorithm is offered.

Key words: software engineering, UML-diagrams, Petri nets, CPN Tools, parallel processes, hyper-threading, system from two connected links.