



ОТКРЫТАЯ НАУКА
издательство

Международный журнал информационных технологий и
энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.896

СПОСОБ ИНТЕРПРЕТАЦИИ БЛОК-СХЕМ АЛГОРИТМОВ В СЕТИ ПЕТРИ ДЛЯ ПРОГРАММИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ

Сорокин Е.В.

Филиал федерального государственного бюджетного образовательного учреждения высшего образования «Национальный исследовательский университет МЭИ» в г. Смоленске, Россия (214013, г. Смоленск, Энергетический проезд, дом 1); e-mail: scorpwork@mail.ru

Настоящая статья посвящена рассмотрению вопросов разработки алгоритмов поведения интеллектуальных агентов, основанной на предварительной интерпретации блок-схем алгоритмов в сети Петри, показана целесообразность и необходимость данного преобразования. Рассмотрены основные задачи над сетями Петри и на их примере представлен анализ возможностей по верификации программного кода. Приведен пример работы интеллектуального агента-сапера, решающего задачу поиска опасных объектов, и показана соответствующая ему сеть Петри.

Ключевые слова: интеллектуальный агент, сеть Петри, алгоритм поведения

METHOD OF INTERPRETATION ALGORITHM FLOWCHART TO PETRI NET FOR PROGRAMMING INTELLIGENT AGENTS

Sorokin E.V.

The Branch of Federal state budgetary educational institution of higher education "National research University Moscow power engineering Institute" in Smolensk, Russia (214013, Smolensk, Energeticheskiy proezd, 1); e-mail: scorpwork@mail.ru

This article is devoted to consideration of questions of development of algorithms of behavior of intelligent agents, based on a preliminary interpretation of the flowcharts of algorithms to Petri nets, the expediency and necessity of this conversion. The main problems on Petri nets and their example presents an analysis of opportunities for program code verification. The work provides an example of an intelligent agent-minesweeper, solving the problem of finding objects is dangerous, and shows the corresponding Petri net.

Key words: intelligent agent, Petri net, behavior algorithm

Разработка алгоритмов поведения интеллектуального агента – работа является сложной задачей, решение которой должно учитывать множество факторов: данные об окружающей среде, технические характеристики аппаратной платформы и глубокое понимание задачи, стоящей перед роботом. Кроме того, разрабатываемые алгоритмы должны быть адаптивными

и обеспечивать работу в независимости от возникающих непредвиденных факторов, влияющих на деятельность робота. Как правило, все особенности способен учесть только профессионал – эксперт в конкретной предметной области. Однако, далеко не всегда он способен запрограммировать робота. Как правило, эта задача возлагается на программиста [1].

На сегодняшний день проектирование и разработка любого программного обеспечения, в том числе программирование интеллектуальных агентов, состоит как минимум из двух этапов: проектирование и программирование. Проектирование осуществляется посредством графического описания будущей системы в виде диаграмм, представленных в той или иной нотации. Программирование робота осуществляется на основе спроектированных графических моделей с использованием различных языков программирования. Как правило, алгоритм работы описывают с помощью блок-схемы. Однако нарисованная диаграмма может иметь множество ошибок и недостатков, приводить к конфликтам в системе.

В связи с этим, пользователю необходимо предоставить возможность программирования робота с помощью простейших операций, не требующих специальных навыков, например путем рисования блок-схем [2]. Но в то же время, использование блок-схемы, бизнес-процесса или иной формы описания алгоритма в чистом виде не дает дальнейших возможностей по применению математического аппарата или моделирования самого процесса, представленного схемами. Предлагается интерпретировать графические диаграммы в сети Петри, модели которых позволяют исследовать работоспособность моделируемых систем, оптимальность их структуры, эффективность процесса их функционирования, а также возможность достижения в процессе функционирования определенных состояний, что в свою очередь позволит сделать выводы о сконструированном экспертом алгоритме поведения робота. Заметим, что алгоритмы, которые в дальнейшем будут прошиваться в программно-аппаратную платформу разрабатываются в ходе моделирования поведения интеллектуальных агентов, поэтому в рамках данной статьи будем считать, что они эквивалентны друг другу.

Сети Петри (СП) и их многочисленные модификации являются одним из классов моделей, неоспоримым достоинством которых является возможность адекватного представления не только структуры сложных организационно-технологических систем и комплексов, но также и логико-временных особенностей процессов и их функционирования, что особенно актуально при моделировании поведения робота. Сети Петри представляют собой математическую модель для представления структуры и анализа динамики функционирования систем в терминах «условие-событие». Эта модель может быть успешно использована для описания так называемых динамических дискретных систем различных классов, таких как: вычислительные процессы и программы, технологические процессы, информационные, экономические, биологические, социальные и технические системы.

Сети Петри и их обобщения являются удобным и мощным средством моделирования асинхронных, параллельных распределительных и недетерминированных процессов, позволяют наглядно представить динамику функционирования систем и составляющих их элементов. Свойство иерархического вложения сетей Петри позволяет рассматривать модели различной степени детализации, обеспечивая тем самым необходимую декомпозицию сложных систем и процессов [6].

При моделировании процессов функционирования дискретных динамических систем, исследование характеристик сетей Петри предполагает решение следующих задач [7].

1. Задача достижимости маркировок: для заданной СП $C = (P, T, I, O, m_0)$ и заданной маркировки m_r установить выполнение условия $m_r \in D(m_0)$.
2. Задача достижимости подмаркировки: для заданной СП $C = (P, T, I, O, m_0)$ и заданной маркировки m_r и подмножества $P' \in P$ определить существует ли достижимая маркировка m_w , компоненты вектора которой с номерами позиций из подмножества P' равны соответствующим компонентам вектора маркировки m_r . Остальные компоненты вектора m_r могут принимать произвольные значения.
3. Задача покрываемости маркировки: для заданной СП $C = (P, T, I, O, m_0)$ и заданной маркировки m_r определить, существует ли достижимая маркировка $m_w \in D(m_0)$, такая что $m_w \geq m_r$.
4. Задача достижимости нуля: для заданной СП $C = (P, T, I, O, m_0)$ определить, достижима ли нулевая маркировка $m = (0, 0, \dots, 0)$ из начальной маркировки m_0 .
5. Задача достижимости нуля в одной позиции: для заданной СП $C = (P, T, I, O, m_0)$ и некоторой фиксированной позиции $p_i \in P$ определить, существует ли достижимая маркировка $m \in D(m_0)$, в которой $m_i = 0$.
6. Задача равенства: для двух СП $C' = (P', T', I', O', m_0')$ и $C'' = (P'', T'', I'', O'', m_0'')$, таких что $P' = P''$, определить, равны ли соответствующие им множества достижимых маркировок $D(m_0') = D(m_0'')$.
7. Задача подмножества: для двух СП $C' = (P', T', I', O', m_0')$ и $C'' = (P'', T'', I'', O'', m_0'')$ определить, является ли одно из множеств достижимых маркировок подмножеством другого.
8. Задача К-ограниченности позиций и СП в целом: для заданной СП $C = (P, T, I, O, m_0)$ и фиксированной позиции $p_i \in P$ определить, существует ли натуральное число К, для которого позиция p_i является К-ограниченной.
9. Задача безопасности позиций и СП в целом: для заданной СП $C = (P, T, I, O, m_0)$ и фиксированной позиции $p_i \in P$ определить, является ли позиция p_i безопасной.
10. Задача устойчивости переходов: для заданной СП $C = (P, T, I, O, m_0)$ и заданного подмножества переходов T' , определить являются ли переходы из подмножества устойчивыми.
11. Задача активности переходов: для заданной СП $C = (P, T, I, O, m_0)$ и заданного подмножества переходов T' определить уровни активности переходов заданного подмножества.
12. Задача сохраняемости СП: для заданной СП $C = (P, T, I, O, m_0)$ определить, является ли она сохраняющей.
13. Задача строгой сохраняемости СП: для заданной СП $C = (P, T, I, O, m_0)$ определить, является ли она строго сохраняющей.

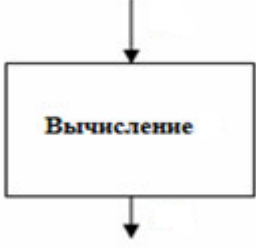
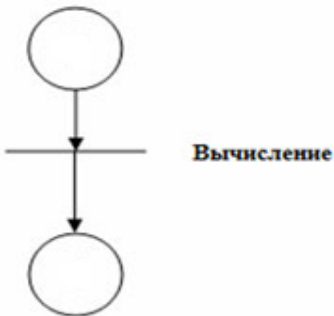

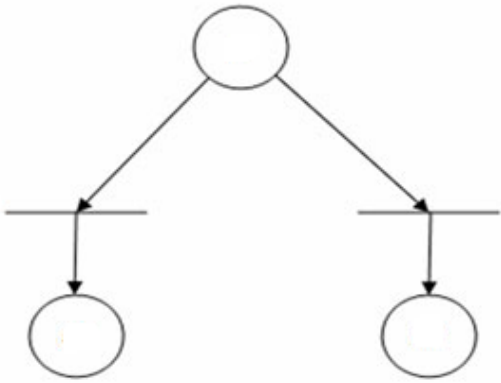
Сконструированный алгоритм работы интеллектуального агента с последующим преобразованием его в сеть Петри, с учетом выше перечисленных задач, позволяет получить сведения о следующих характеристиках:

- о количестве процессов в системе;
- наличии взаимоблокировок (преимущественно задача устойчивости переходов, задача достижимости нуля);

- наличии невыполнимых операций (задачи достижимости маркировок, подмаркировок и задача покрываемости);
- количестве циклов, которые при определенных ситуациях могут стать причиной заикливания (задачи устойчивости и достижимости нуля).

Как было сказано ранее, наиболее распространенным подходом к описанию алгоритма является рисование блок-схемы. Блок-схема во многом подобна сети Петри и представима в виде узлов двух типов (принятия решения, показанные ромбами, и вычисления, показанные прямоугольниками) и дуг между ними. Одним из способов выполнения блок-схемы является введение фишки, которая представляет текущую инструкцию. По мере выполнения инструкций фишка передвигается по блок-схеме. Перевод блок-схемы в сеть Петри заменяет узлы блок-схемы на переходы сети Петри, а дуги блок-схемы – на позиции сети Петри. Каждая дуга блок-схемы соответствует точно одной позиции в сети Петри. Узлы блок-схемы представляются по-разному в зависимости от типа узла: вычисления или принятия решения. Фишка, находящаяся в позиции, означает, что счетчик команд установлен на готовность выполнения следующей инструкции. Каждая позиция имеет единственный выходной переход, за исключением позиции, которая имеет по два выходных перехода, соответствующих истинному и ложному значению предиката [3]. Для интерпретации сети Петри необходимо отображать каждый переход. Следует также отметить, что переходы для вычислений имеют по одному входу и выходу. В таблице 1 иллюстрируются оба способа перевода [4,5].

Таблица 1 – Перевод узлов вычисления и принятия решения в блок-схеме в переходы сети Петри

№ п/п	Элемент блок-схемы	Вид композиции в виде сети Петри
1	2	3
1		
2		

На рисунке 1 показан пример блок-схемы алгоритма работы интеллектуального агента-сапера в процессе поиска мин на поле. Начало работы осуществляется сразу за поступлением соответствующей команды, после которой включаются нужные сенсоры (метало-искатель, лампы, дальномеры (блок 2)), затем с некоторой скоростью агент начинает движение, сканируя перед собой местность. Блок 4 на рисунке 1 отвечает за определение наличия металла перед роботом, при этом в любой момент извне может поступить команда окончания работы (блок 5), при которой робот останавливается и заканчивает работу. При обнаружении искомого элемента робот останавливается, передает координаты опасного объекта и объезжает его (блоки 6,7,8), продолжая движение и основную работу.

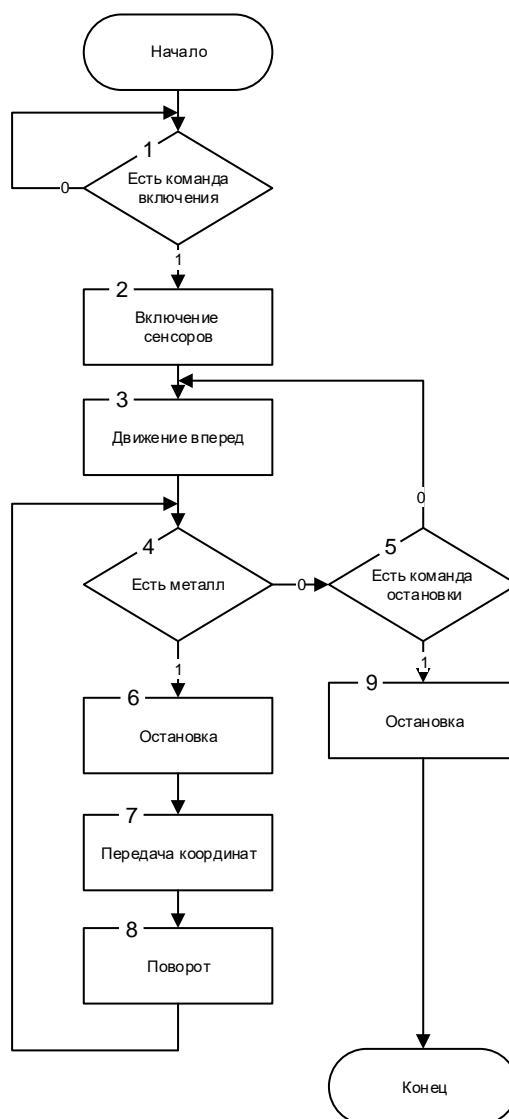


Рисунок 1 – Блок-схема работы интеллектуального агента по поиску мин

На рисунке 2 показана соответствующая блок-схеме агента-сапера (рисунок 1) сеть Петри, интерпретируемая по описанным выше правилам.

Представленная на рисунке 2 сеть Петри моделирует поведение робота при решении поставленной перед ним задачи, наглядно показывает динамику функционирования системы. Введение временных и условных ограничений к переходам сети позволит уточнить процесс функционирования модели и приблизит ее к реальной работе аппаратной платформы.

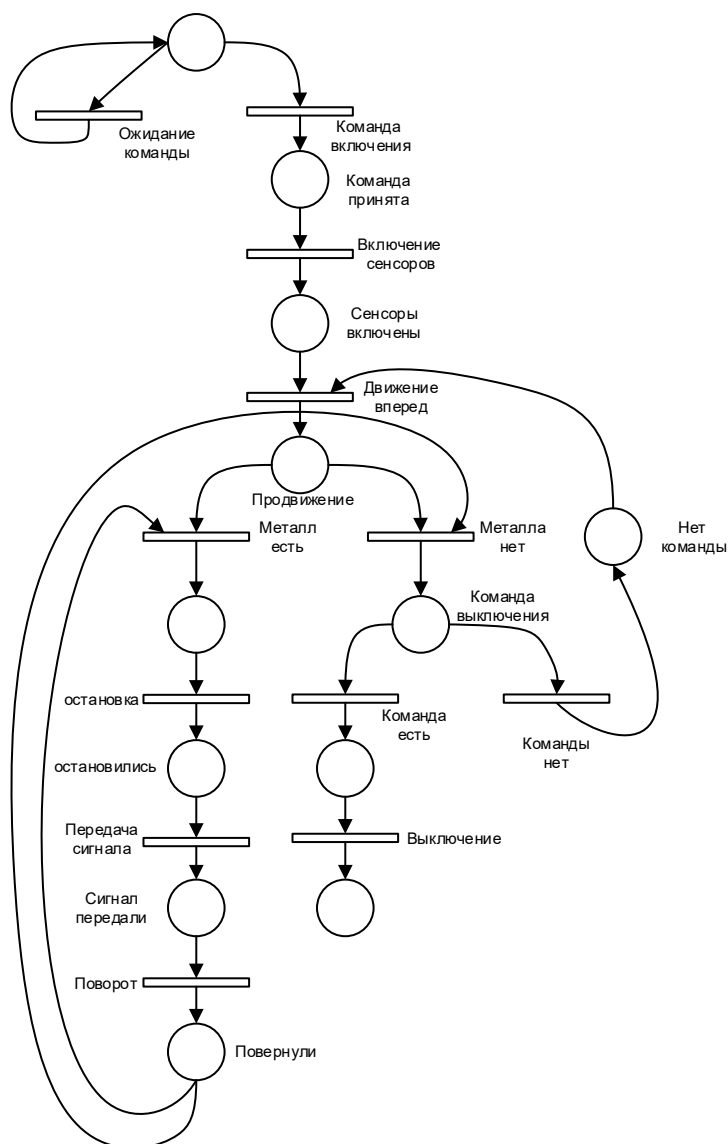


Рисунок 2 - Сеть Петри алгоритма поведения интеллектуального агента по поиску мин на поле

Предложенный способ интерпретации блок-схем алгоритмов в сеть Петри лежит в основе способа генерации алгоритмов поведения интеллектуальных агентов и предоставляет возможности по верификации введенной пользователем схемы, позволяет провести ее глубокий анализ, а также может служить основой для дальнейшего усложнения сети с целью оптимизации выходного алгоритма для интеллектуального агента.

Список литературы

1. Сорокин Е.В. Визуальной среда конструирования алгоритмов поведения интеллектуального агента // Сборник трудов 13-ой международной научно-технической конференции студентов и аспирантов «Информационные технологии, энергетика и экономика» 14-15 апреля 2016 года. Филиал МЭИ в г. Смоленске. Смоленск. 2016. Т. 1. С. 382-385.

2. Сеньков, А.В. Способ и средства визуального конструирования автоматизированных информационных систем [Электронный ресурс]/ А.В. Сеньков, А. Г.Забурдаев// Математическая морфология. Электронный математический и медико-биологический журнал. – Т. 9. – Вып. 4. – 2010. – URL: Режим доступа к журн.: <http://www.smolensk.ru/user/sgma/MMORPH/N-28-html/TITL-28.htm>
3. Котов В.Е. Сети Петри. М.:Наука, Гл. ред. физ.-мат. лит., 1984. 160с.
4. Рудаков И.В., Пащенко А.В. Программный комплекс верификации алгоритмов программного обеспечения с помощью иерархических сетей Петри. СГТУ им. Н.Э. Баумана, Москва 105005, Россия. 10с.
5. Норенков И.П. Основы автоматизированного проектирования: учеб. для вузов. М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. 306 с.
6. Питерсон Дж. Теория сетей Петри и моделирование систем: пер. с англ. М.: Мир. 1984. 264с.
7. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2005. – 736 с.

References

1. Sorokin E.V. Visual environment design of algorithms of behavior of an intelligent agent // Proceedings of the 13th international scientific and technical conference of students and postgraduate students "Information technologies, energy and the economy" 14-15 April 2016 year. Branch MPEI in Smolensk. Smolensk. 2016 T.1 P.382-385
 2. Senkov, A.V. Spособ i sredstva vizual'nogo konstruirovaniya avtomatizirovannyh informacionnyh sistem [Jelektronnyj resurs]/ A.V. Sen'kov, A. G.Zaburdaev// Matematicheskaja morfologija. Jelektronnyj matematicheskij i mediko-biologicheskij zhurnal. – Т. 9. – Vyp. 4. – 2010. – URL: Rezhim dostupa k zhurn.: <http://www.smolensk.ru/user/sgma/MMORPH/N-28-html/TITL-28.htm>
 3. Kotov V. E. Petri Nets. M.:Nauka, GL. ed. Fiz.-Mat. lit., 1984. 160P
 4. Rudakov I. V., A. V. Pashenkov Software verification software algorithms using hierarchical Petri nets. SGTU them. N. Uh. Bauman, Moscow 105005, Russia. 10P
 5. Norenkov I. P. fundamentals of CAD: proc. for higher education institutions. M.: Izd-vo MGTU im. N. Uh. Bauman, 2002. 306P
 6. Peterson George. The theory of Petri nets and modeling of systems: per. from English. M.: Mir. 1984. – 264P
 7. The Leonenko A.V. Fuzzy modeling in MATLAB and fuzzyTECH. – SPb.: BHV-Petersburg, 2005. – 736 p.
-