

ИНФОРМАТИКА,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И УПРАВЛЕНИЕ

INFORMATICS,
COMPUTER ENGINEERING
AND CONTROL

УДК004.4'22

DOI: 10.17212/1814-1196-2018-4-91-100

Использование сетей Петри при проектировании архитектуры программного продукта для анализа данных с помощью нейронных сетей*

В.А. ХАРАХИНОВ^a, С.С. СОСИНСКАЯ^b

664074, РФ, г. Иркутск, ул. Лермонтова, 83, Иркутский национальный исследовательский технический университет

^a tes4obse@mail.ru ^b sosinskaya@mail.ru

В статье рассматривается этап проектирования, разрабатываемого авторами, программного комплекса, реализующего процессы кластеризации и классификации данных на основе искусственных нейронных сетей с предварительным сокращением размерности (редуцированием) пространства признаков. Этот этап является одним из важнейших при разработке программного продукта. Практически все современные программные комплексы характеризуются большим разнообразием взаимодействующих программных модулей, что, в свою очередь, увеличивает сложность на этапе проектирования современных программ. На этом этапе разработчики программных продуктов используют различные методы. В статье рассмотрен известный способ моделирования систем математическим аппаратом сетей Петри, который получил широкое применение в самых разных предметных областях. Построена сеть Петри для проектируемого программного комплекса, содержащая 48 вершин.

Проведен анализ построенной сети Петри. Он дает представление о поведении проектируемого программного комплекса. Также качественный анализ сети способствует внесению изменений в архитектуру программного комплекса на этапе проектирования.

Анализ сети производился в программной среде CPNTools версии 4.0.1. Данная среда обладает автоматизированными средствами анализа и генерирует отчеты о тупиковых состояниях системы, мертвых переходах. Анализ, проводимый в этой среде, по своей сути решает одну из основных задач теории сетей Петри – задачу достижимости.

Результатом анализа являются оценка достижимых маркировок и выявление мертвых и тупиковых состояний, приводящих к прекращению работы системы, заикливание и выявлению сценариев, не участвующих в работе системы. Одним из ключевых результатов анализа стало дерево достижимости. Как показал анализ, в спроектированной сети отсутствуют мертвые переходы, что сигнализирует о корректно спроектированной архитектуре программного продукта.

Ключевые слова: моделирование, сеть Петри, анализ данных, классификация, кластерный анализ, нейронная сеть, редукция данных, факторный анализ, разработка ПО

* Статья получена 06 августа 2018 г.

ВВЕДЕНИЕ

При проектировании алгоритмов и создании программных систем необходим их анализ с целью проверки непротиворечивости и достижения оптимальности функционирования. Часто для этой цели используется аппарат сетей Петри, которые получили широкое применение в самых разных предметных областях: моделированию аппаратом сетей Петри может быть подвергнута почти любая система, алгоритм или процесс. Это стало возможным благодаря интенсивному развитию сетей Петри различных модификаций.

Анализ проектируемого программного продукта состоит в оценке достижимых маркировок и выявлении мертвых и тупиковых состояний, приводящих к прекращению работы системы, заикливанию и выявлению сценариев, не участвующих в работе системы, что способствует внесению изменений в процесс проектирования.

В статье показан процесс построения сети Петри проектируемого программного продукта для кластеризации и классификации данных с предварительным сокращением размерности (редуцированием) пространства признаков и ее анализ.

1. СЕТЬ ПЕТРИ

В самом общем случае сетью Петри называется совокупность множеств

$$C = \{P, T, I, O\},$$

где P – конечное множество, элементы которого называются позициями; T – конечное множество, элементы которого называются переходами; I – множество входных функций, $I: T \rightarrow P$; O – множество выходных функций, $O: T \rightarrow P$.

Для моделирования сети необходимо иметь маркировку сети μ . Маркировка μ определяется как n -вектор $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, где $n = |P|$ и каждое $\mu_i \in \mathbb{N}$, $i = 1, \dots, n$. Вектор μ определяет для каждой позиции p_i сети Петри количество меток в этой позиции [1].

Маркированная сеть Петри может быть записана в виде: $M = (C, \mu)$ или $M = (P, T, I, O, \mu)$.

Сеть Петри моделируется посредством запуска переходов. Переход запускается удалением меток из его входных позиций и образованием новых меток, помещаемых в его выходные позиции. Следовательно, при моделировании сети маркировка сети будет изменяться.

Переход может запускаться только в том случае, если он разрешен. Переход $t_j \in T$ в маркированной сети Петри $C = (P, T, I, O)$ с маркировкой μ разрешен, если для всех $p_i \in P$:

$$\mu(p_i) \geq \#(p_i, I(t_j)).$$

Переход запускается удалением всех разрешающих меток из его входных позиций и последующим помещением в каждую из его выходных позиций по одной метке для каждой дуги [1].

Простое представление системы сетью Петри основано на двух основополагающих понятиях: событиях и условиях. События – это действия, имеющие место в системе. Условие есть предикат либо логическое описание состояния системы. В сети Петри условия моделируются позициями, события – переходами. Возникновение события равносильно запуску соответствующего перехода [2].

Для моделирования процесса или системы аппаратом сетей Петри необходимо определить ее вершины – позиции и переходы.

2. ОПИСАНИЕ ПРОЕКТИРУЕМОГО ПРОГРАММНОГО ПРОДУКТА

Сеть Петри будет отображать архитектуру разработанного авторами программного комплекса [3], который реализует анализ данных путем использования различных нейронных сетей.

Для анализа данных в программном комплексе используются такие нейронные сети, как многослойный персептрон, слой и карта Кохонена, сеть векторного квантования, сеть распознавания [4].

Однако, как уже говорилось ранее, поддерживается процесс редукции данных, который реализован посредством использования факторного анализа [5, 6] и применения автокодировщика (нейронной сети, способной обучаться без учителя, при помощи метода обратного распространения ошибки) [7].

Резюмируя вышеперечисленное, можно выделить следующие задачи, которые программный комплекс должен обеспечивать:

- 1) выбор и загрузка файла с данными для анализа;
- 2) проведение редукции данных (опционально);
- 3) выбор типа нейронной сети и параметров обучения для нее;
- 4) обучение выбранной сети на загруженных данных;
- 5) создание тестирующей выборки из загруженных данных;
- 6) проведение анализа данных при помощи обученной сети с тестирующей выборкой;
- 7) отображение результатов анализа;
- 8) формирование отчетных документов в форматах MAT и XLS.

Исходя из вышеперечисленных условий были определены вершины сети Петри и связи между ними.

Графически позицию принято обозначать овалом, переход – прямоугольником. На рис. 1 представлена сеть Петри для проектируемого программного комплекса. Она содержит 48 вершин. В начальном состоянии сети лишь одна позиция сети «Выбор в главном меню программы» содержит метку.

Процесс моделирования системы аппаратом сетей Петри также включает один важный этап – анализ спроектированной сети. Важность данного этапа заключается в выявлении тупиковых состояний, мертвых переходов или заикливаниях в ходе моделирования сети [8].

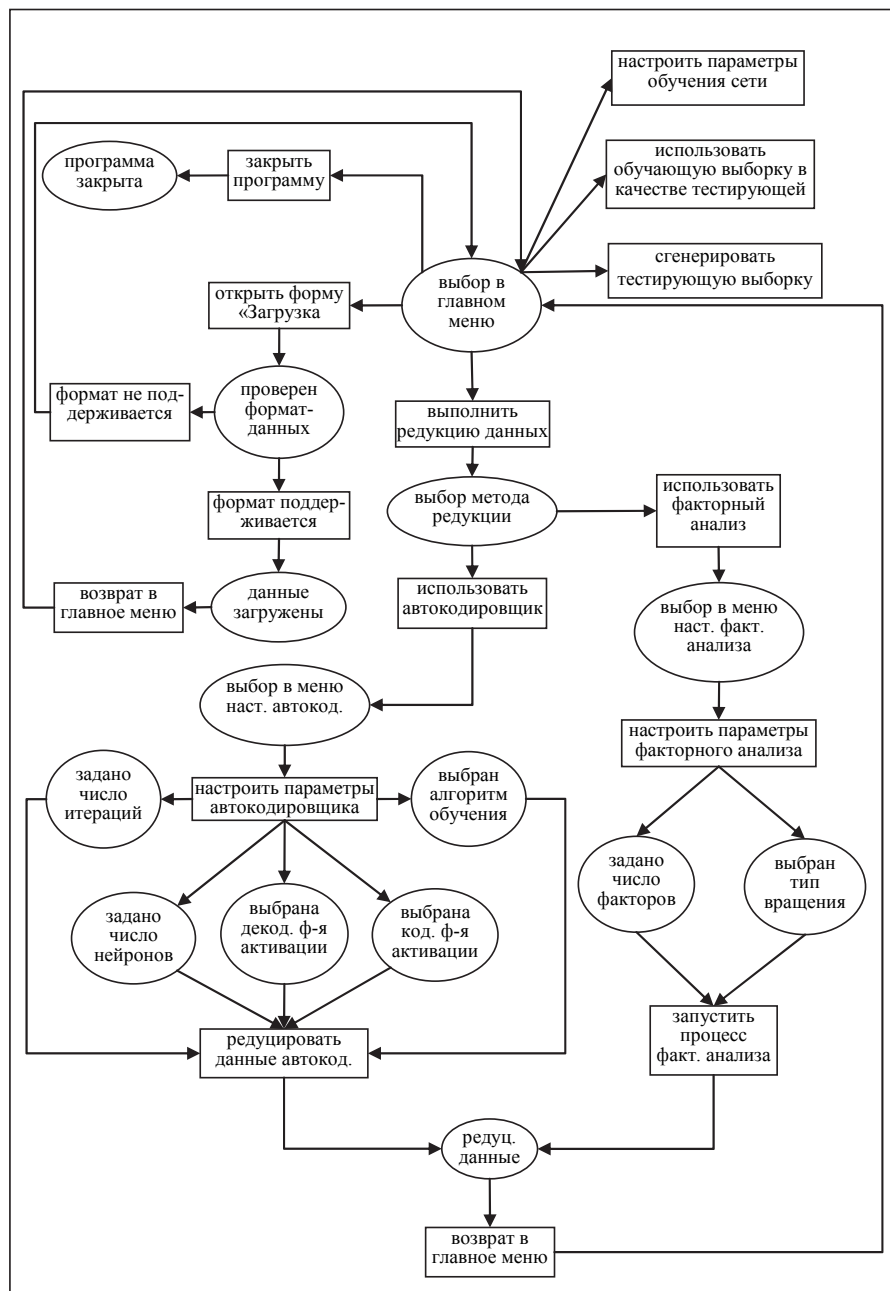


Рис. 1. Сеть Петри для проектируемого программного комплекса (часть 1)

Fig. 1. The Petri net for the designed software complex (part 1)

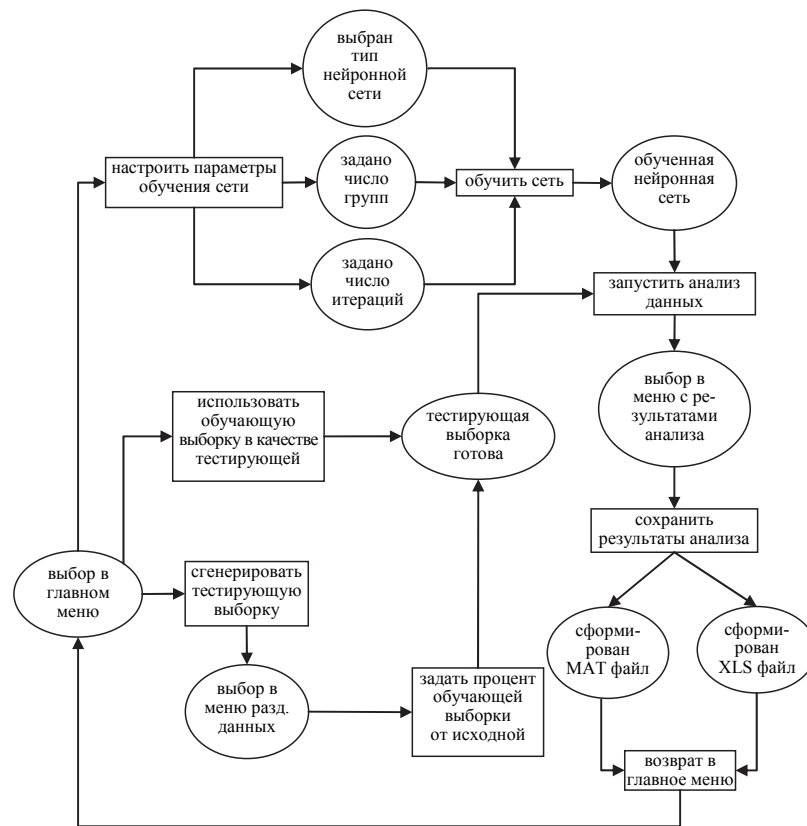


Рис. 2. Сеть Петри для проектируемого программного комплекса (часть 2)

Fig. 2. The Petri net for the designed software complex (part 2)

3. АНАЛИЗ СПРОЕКТИРОВАННОЙ СЕТИ ПЕТРИ

Анализ сети Петри дает представление о поведении проектируемого программного комплекса.

Анализ сети производился в программной среде CPNTools версии 4.0.1 [9]. Данная среда обладает автоматизированными средствами анализа и генерирует отчеты о тупиковых состояниях системы и мертвых переходах. Анализ, проводимый в этой среде, по своей сути решает одну из основных задач теории сетей Петри – задачу достижимости.

Формально задача достижимости состоит в следующем: для сети Петри с начальной маркировкой μ_0 и заданной маркировкой μ установить справедливость включения $\mu \in R(C, \mu_0)$. Иными словами, требуется выяснить, существует ли допустимая последовательность срабатываний переходов, переводящая сеть Петри из начальной маркировки в заданную [10].

В отличие от многих других программных продуктов, реализующих логику математического аппарата сетей Петри и обеспечивающих пользовательский интерфейс для работы с ними, в приведенной версии среды CPNTools поддерживается автоматизированное построение деревьев достижимости [11].

В среде CPNTools была построена сеть, идентичная приведенной на рис. 1, и проведен ее анализ. Рисунок 2 отображает полученное в результате анализа дерево достижимости.

Узлы дерева в среде CPNTools отображаются в виде рис. 3.



Рис. 3. Отображение узла дерева в среде CPN Tools

Fig. 3. A tree node in the CPN Tools software environment

На рисунке N – числовой номер узла дерева, IN – число входных дуг (ветвей), OUT – число выходных дуг.

Построение дерева достижимости начинается с первого (корневого) узла дерева (на рис. 4 узел дерева имеет числовой номер «1»). Можно заметить, что число узлов дерева заметно меньше вершин графа (рис. 1 и 2). Причиной этому послужили активации таких переходов, как «настроить параметры автокодировщика», «настроить параметры обучения сети», «настроить параметры факторного анализа». В результате активации приведенных переходов происходил перенос меток из входной позиции в несколько выходных позиций. Однако на дереве достижимости подобное увеличение меток в позициях отображается одним узлом.

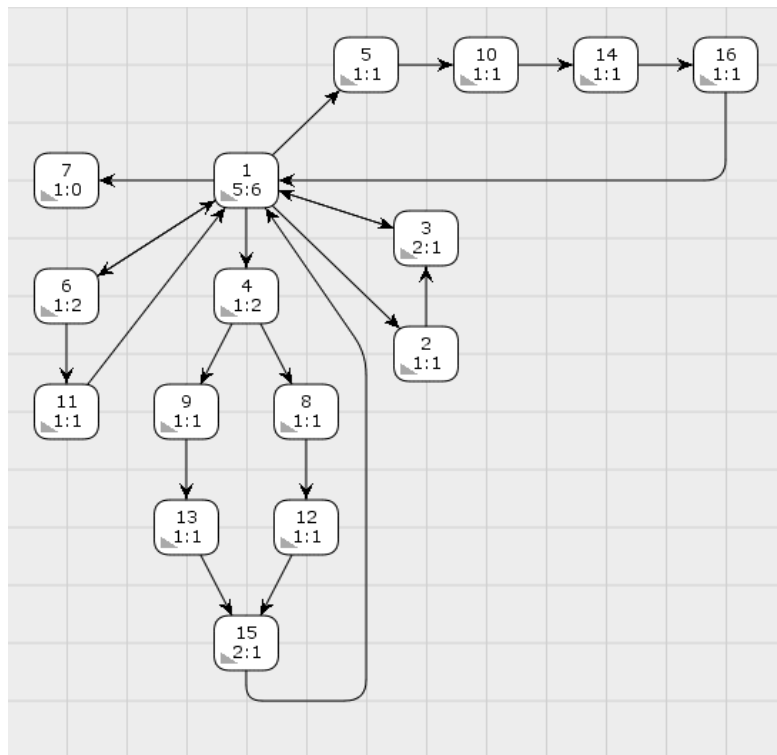


Рис. 4. Дерево достижимости сети проектируемого программного комплекса

Fig. 4. The reachability graph for the designed software complex

Как показал анализ, в спроектированной сети отсутствуют мертвые переходы, что сигнализирует о корректно спроектированной архитектуре программного продукта [12].

ЗАКЛЮЧЕНИЕ

В данной работе описана возможность использования математического аппарата сетей Петри на стадии проектирования программного продукта, что позволяет выявить неточности в проектировании архитектуры программных продуктов. Описанный в статье программный комплекс был реализован в среде MATLAB согласно спроектированной архитектуре [3]. После выполнения программной реализации был проведен ряд исследований по анализу данных [13–15] с использованием данного программного продукта.

СПИСОК ЛИТЕРАТУРЫ

1. Котов В.Е. Сети Петри. – М.: Наука, 1984. – 160 с.
2. Reisig W. Petri nets: an introduction. – Berlin; New York: Springer-Verlag, 1985. – 172 p.
3. Программный комплекс «Анализ экспериментальных данных на основе нейронных сетей»: свидетельство о государственной регистрации программы для ЭВМ / В.А. Харахинов, С.С. Сосинская. – № 2017618294; зарег. 04.07.2017.
4. Хайкин С. Нейронные сети: полный курс. – 2-е изд. – М.: Вильямс, 2006. – 1104 с.
5. Харман Г. Современный факторный анализ. – М.: Статистика, 1972. – 484 с.
6. Лоули Д., Мансвелл А. Факторный анализ как статистический метод. – М.: Мир, 1967. – 141 с.
7. Hinton G.E., Zemel R.S. Autoencoders, minimum description length and Helmholtz free energy // *Advances in Neural Information Processing Systems*. – 1994. – Vol. 6. – P. 3–10.
8. Марков А.В., Воевода А.А. Анализ сетей Петри при помощи деревьев достижимости // *Сборник научных трудов НГТУ*. – 2013. – № 1. – С. 78–95.
9. Официальный сайт разработчиков среды CPN Tools [Электронный ресурс]. – URL: <http://cpntools.org/> (дата обращения: 11.12.2018).
10. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.
11. Романников Д.О., Марков А.В. Об использовании программного пакета CPN Tools для анализа сетей Петри // *Сборник научных трудов НГТУ*. – 2012. – № 2. – С. 105–116.
12. Girault C., Valk R. Petri nets for systems engineering: a guide to modeling, verification and applications. – New York: Springer, 2003. – 628 p.
13. Харахинов В.А., Сосинская С.С. Исследование способов кластеризации деталей машиностроения на основе нейронных сетей // *Программная инженерия*. – 2017. – № 4. – С. 170–176.
14. Харахинов В.А., Сосинская С.С. Влияние сокращения размерности пространства признаков на результаты классификации листьев различных видов растений // *Программная инженерия*. – 2018. – № 2. – С. 82–90.
15. Харахинов В.А. Генетический алгоритм как альтернатива обучения слоя Кохонена // *Информационные технологии*. – 2018. – № 10. – С. 642–648.

Харахинов Владимир Александрович, аспирант кафедры вычислительной техники Иркутского национального исследовательского технического университета. Основное направление научных исследований – искусственный интеллект, распознавание образов, параллельные вычисления. E-mail: tes4obse@mail.ru

Сосинская Софья Семеновна, кандидат технических наук, профессор кафедры вычислительной техники Иркутского национального исследовательского технического университета. Основное направление научных исследований – искусственный интеллект, распознавание образов, параллельные вычисления. Имеет более 70 печатных работ и учебных пособий. E-mail: sosinskaya@mail.ru

Kharakhinov Vladimir Alexandrovich, graduate student at the Department of Computing Engineering in INRTU. The main field of his research is artificial intelligence, pattern recognition, and parallel computing. E-mail: tes4obse@mail.ru

Sosinskaya Sofya Semenovna, PhD, professor, Department of Computing Engineering, INRTU. Her research interests are currently focused on artificial intelligence, pattern recognition, and parallel computing. She has more than 70 publications and teaching manuals. E-mail: sosinskaya@mail.ru

DOI: 10.17212/1814-1196-2018-4-91-100

The use of Petri nets for the software product design to analyze data using neural networks*

V.A. KHARAKHINOV^a, S.S. SOSINSKAYA^b

Irkutsk National Research Technical University, 83, Lermontov Street, Irkutsk, 664074, Russian Federation

^a tes4obse@mail.ru ^b sosinskaya@mail.ru

Abstract

This paper considers the design stage of the software development process to carry out cluster analysis and classification based on artificial neural networks with a preliminary reduction of attribute space dimension also based on artificial neural networks and additionally on factor analysis. The design stage is one of the most important software development stages. Most modern software systems contain a great variety of cooperating program modules which in its turn increases the complexity of the design process. At this stage software developers use various methods. This article describes a well-known method of system modeling based on the Petri net which has been widely used in various subject areas. A Petri net with 48 vertexes was created for the developed software complex.

The realized Petri net is analyzed. The analysis provides insights into the behavior of the developed program.

The Petri net analysis was performed in the CPN Tools software environment (version 4.0.1). This environment has automated analysis tools and generates reports on deadlock states and deadlock state transitions. The analysis conducted in this environment, inherently solves one of the main problems of the Petri nets theory, namely the reachability problem.

High-quality analysis contributes to the adjustments in the software architecture at the design stage.

The analysis results are the estimation of reachable markings, identification of deadlock states that lead to a system's shutdown, looping and detection of scripts that are not participating in the system. One of the key results is the reachability graph. The analysis has shown that there are no deadlock transitions, which means that the software architecture as been designed correctly.

Keywords: modeling, Petri nets, data analysis, classification, cluster analysis, neural network, data reduction, factor analysis, software development

* Received 06 August 2018.

REFERENCES

1. Kotov V.E. *Seti Petri* [Petri nets]. Moscow, Nauka Publ., 1984. 160 p.
2. Reisig W. *Petri nets: an introduction*. Berlin, New York, Springer-Verlag, 1985. 172 p.
3. Kharakhinov V.A., Sosinskaya S.S. *Programmnyi kompleks "Analiz eksperimental'nykh dannykh na osnove neironnykh setei"*. The certificate on official registration of the computer program. No. 2017618294, 2017. (In Russian, unpublished).
4. Haykin S. *Neural networks: a comprehensive foundation*. 2nd ed. Upper Saddle River, NJ, Prentice Hall, 1999 (Russ. ed.: Khaikin S. *Neironnye seti: polnyi kurs*. 2nd ed. Moscow, Villiams Publ., 2006. 1104 p.).
5. Harman H.H. *Modern factor analysis*. 2d ed., rev. Chicago, University of Chicago Press, 1967 (Russ. ed.: Kharman G. *Sovremennyi faktornyi analiz*. Moscow, Statistika Publ., 1972. 484 p.).
6. Lawley D.N., Maxwell A.E. *Factor analysis as a statistical method*. London, Butterworths, 1963 (Russ. ed.: Louli D., Mansvell A. *Faktornyi analiz kak statisticheskii metod*. Moscow, Mir Publ., 1967. 141 p.).
7. Hinton G.E., Zemel R.S. Autoencoders, minimum description length and Helmholtz free energy. *Advances in Neural Information Processing Systems*, 1994, vol. 6, pp. 3–10.
8. Markov A.V., Voevoda A.A. Analiz setei Petri pri pomoshchi derev'ev dostizhimosti [Analysis of Petri nets by trees reachable]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2013, no. 1, pp. 78–95.
9. *CPN Tools*: website. Available at: <http://cpntools.org/> (accessed 11.12.2018).
10. Peterson J.L. *Petri net theory and the modeling of systems*. Englewood Cliffs, NJ, Prentice-Hall, 1981 (Russ. ed.: Piterson Dzh. *Teoriya setei Petri i modelirovanie sistem*. Moscow, Mir Publ., 1984. 264 p.).
11. Romannikov D.O., Markov A.V. Ob ispol'zovanii programmnoho paketa CPN Tools dlya analiza setei Petri [The review of works describing sharing UML diagrams and Petri nets at a design stage of software]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2012, no. 2, pp. 105–116.
12. Girault C., Valk R. *Petri nets for systems engineering: a guide to modeling, verification and applications*. New York, Springer, 2003. 628 p.
13. Kharakhinov V.A., Sosinskaya S.S. Issledovanie sposobov klasterizatsii detalei mashinostroeniya na osnove neironnykh setei [The visualization methods for cluster analysis results of mechanical engineering components based on neural network]. *Programmnaya inzheneriya – Software Engineering*, 2017, no. 4, pp. 170–176.
14. Kharakhinov V.A., Sosinskaya S.S. Vliyanie sokrashcheniya razmernosti prostranstva priznakov na rezul'taty klassifikatsii list'ev razlichnykh vidov rastenii [The effect of dimension reducing on classification results of leaves of various plant species]. *Programmnaya inzheneriya – Software Engineering*, 2018, no. 2, pp. 82–90.
15. Kharakhinov V.A. Geneticheskii algoritm kak al'ternativa obucheniya sloya Kokhonena [The genetic algorithm as the alternative method for training Kohonen layer]. *Informatsionnye tekhnologii – Information Technologies*, 2018, no. 10, pp. 642–648.

Для цитирования:

Харахинов В.А., Сосинская С.С. Использование сетей Петри при проектировании архитектуры программного продукта для анализа данных с помощью нейронных сетей // Научный вестник НГТУ. – 2018. – № 4 (73). – С. 91–100. – doi: 10.17212/1814-1196-2018-4-91-100.

For citation:

Kharakhinov V.A., Sosinskaya S.S. Ispol'zovanie setei Petri pri proektirovanii arkhitektury programmno produkta dlya analiza dannykh s pomoshch'yu neironnykh setei [The use of Petri nets for the software product design to analyze data using neural networks]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2018, no. 4 (73), pp. 91–100. doi: 10.17212/1814-1196-2018-4-91-100.