

STOCHASTIC PETRI NETS SOFTWARE TOOLS

A. V. Bystrov, I. B. Virbitskaite, E. S. Oshevskaya

A. P. Ershov Institute of Informatics Systems,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2024-2-32-57

EDN: KNBRYZ

The behavior of a wide variety of systems, biochemical, transport, industrial, software, and so on, is inherently parallel, non-deterministic, and stochastic. The study and design of these systems requires the use of models that take into account all these aspects, as well as appropriate software tools. Stochastic Petri nets and their various extensions are successfully used as such models. They combine the clarity and intuitiveness of the graphical representation with well-developed mathematical and algorithmic apparatus of analysis. These models allow us to study not only qualitative but also quantitative properties of systems, such as bandwidth, reliability, waiting time, etc. Software tools that support the construction, modification and analysis of system models based on various variants of stochastic Petri nets have already been developed and continue to appear.

This paper provides a detailed overview of several such multiplatform software tools, namely, GreatSPN, ORIS, PetriNuts, TimeNet and PIPE2 that are available on the Internet, and got recognized by users. The introduction, informally, but with proper references to the literature, gives the basic concepts, defines the classes of Petri nets and terms used later. Then, for each of the software tools, its structure, features and peculiarities are considered. The tools are then compared in terms of their functional and performance analysis capabilities, and recommendations to users on how to use the tools depending on what type of stochastic models need to be investigated are discussed. The main purpose of the paper is to facilitate the researcher and engineer in selecting the most appropriate modeling and analysis tool for the task at hand.

Key words: stochastic Petri nets, modelling, simulation, performance analysis, Petri net tools.

References

1. Reisig W. Petri Nets: An Introduction. V. 4. Springer, 1985. (EATCS Monographs on Theoretical Computer Science), DOI: 10.1007/978-3-642-69968-9.
2. Boyer M., Roux O. On the Compared Expressiveness of Arc, Place and Transition Time Petri Nets // *Fundamenta Informaticae*. 2008. Jan. V. 88. P. 225–249.
3. Berthomieu B., Diaz M. Modeling and verification of time dependent systems using time Petri nets // *IEEE Transactions on Software Engineering*. 1991. Mar. V. 17, N 3. P. 259–273. DOI: 10.1109/32.75415.
4. Molloy M. Performance Analysis Using Stochastic Petri Nets // *IEEE Trans. Computers*. 1982. V. 31, N 9. P. 913–917. DOI: 10.1109/TC.1982.1676110.
5. Vicario E., Sassoli L., Carnevali L. Using Stochastic State Classes in Quantitative Evaluation of Dense-Time Reactive Systems // *IEEE Trans. Software Eng.* 2009. V. 35, N 5. P. 703–719. DOI: 10.1109/TSE.2009.36.
6. Wang J. Stochastic Timed Petri Nets and Stochastic Petri Nets // *Timed Petri Nets: Theory and Application*. Boston, MA : Springer US, 1998. P. 125–153. DOI: 10.1007/978-1-4615-5537-7_5.

7. Ajmone Marsan M. et al. An introduction to generalized stochastic Petri nets // *Microelectronics Reliability*. 1991. Jan. V. 31, N 4. P. 699–725. DOI: 10.1016/0026-2714(91)90010-5.
8. Ajmone Marsan M., Chiola G. On Petri nets with deterministic and exponentially distributed Bring times // *Advances in Petri Nets 1987*, covers the 7th European Workshop on Applications and Theory of Petri Nets, Oxford, UK, June 1986. V. 266 / ed. by G. Rozenberg. Springer, 1986. P. 132–145. (Lecture Notes in Computer Science), DOI: 10.1007/3-540-18086-9_23.
9. Dugan J. et al. Extended Stochastic Petri Nets: Applications and Analysis // *Performance '84*, Proceedings of the Tenth International Symposium on Computer Performance Modelling, Measurement and Evaluation / ed. by E. Gelenbe. North-Holland, 1984. P. 507–519.
10. Ajmone Marsan M. et al. The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets // *IEEE Trans. Software Eng.* 1989. V. 15, N 7. P. 832–846. DOI: 10.1109/32.29483.
11. German R., Lindemann C. Analysis of stochastic Petri nets by the method of supplementary variables // *Performance Evaluation*. 1994. May. V. 20, N 1–3. P. 317–335. DOI: 10.1016/0166-5316(94)90020-5.
12. Stokasticheskie seti Petri — formalizm dlya modelirovaniya i analiza proizvoditel'nosti vychislitel'nykh processov // *Sistemnaya Informatika*. Novosibirsk, 2004. S. 135–193 (In Russian).
13. German R. Performance analysis of communication systems — modelling with non- Markovian stochastic Petri nets : Modeling with Non-Markovian Stochastic Petri Nets. Wiley, 2000. P. 456.
14. Biagi M. et al. Exploiting Non-deterministic Analysis in the Integration of Transient Solution Techniques for Markov Regenerative Processes // *Quantitative Evaluation of Systems*. Springer International Publishing, 2017. P. 20–35. DOI: 10.1007/978-3-319-66335-7_2.
15. Martina S. et al. Performance Evaluation of Fischer's Protocol through SteadyState Analysis of Markov Regenerative Processes // *IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 09/2016. P. 355–360. DOI: 10.1109/MASCOTS.2016.72.
16. Horvath A. et al. Transient analysis of non-Markovian models using stochastic state classes // *Performance Evaluation*. 2012. V. 69, N 7/8. P. 315–335. DOI: 10.1016/j.peva.2011.11.002.
17. Amparore E. Stochastic Modelling and Evaluation Using GreatSPN // *ACM- SIGMETRICS Performance Evaluation Review*. New York, NY, USA, 2022. June. V. 49, N 4. P. 87–91. DOI: 10.1145/3543146.3543165.
18. Amparore E. et al. Years of GreatSPN // *Principles of Performance and Reliability Modeling and Evaluation: Essays in Honor of Kishor Trivedi on his 70th Birthday* / ed. by L. Fiondella, A. Puliafito. Cham : Springer International Publishing, 2016. P. 227–254. DOI: 10.1007/978-3-319-30599-8-9.
19. K. J., Kristensen L. Coloured Petri Nets. Springer Berlin Heidelberg, 2009. DOI: 10.1007/b95112.
20. ISO/IEC. Software and Systems Engineering - High-level Petri Nets, Part 2: Transfer Format, International Standard ISO/IEC 15909, February 2011.
21. Kindler E. The Petri Net Markup Language and ISO/IEC 15909-2: Concepts, Status, and Future Directions // *Tagungsband Entwurf komplexer Automatisierungssysteme EKA*. 2006. P. 35–55.
22. Clarke E., Emerson E. Design and synthesis of synchronization skeletons using branching time temporal logic // *Logics of Programs* / ed. by D. Kozen. Springer Berlin Heidelberg, 1982. P. 52–71.
23. Deharbe D. A Tutorial Introduction to Symbolic Model Checking // *Logic for Concurrency and Synchronisation* / ed. by R. de Queiroz. Dordrecht : Springer Netherlands, 2003. P. 215–237. DOI: 10.1007/0-306-48088-3_5.
24. Beccuti M., Franceschinis G., Haddad S. Markov Decision Petri Net and Markov Decision Well-Formed Net Formalisms // *Petri Nets and Other Models of Concurrency — ICATPN 2007* / ed. by J. Kleijn, A. Yakovlev. Springer Berlin Heidelberg, 2007. P. 43–62. DOI: 10.1007/978-3-540-73094-1_6.

25. Emerson E., Sistla A. Symmetry and model checking // *Formal Methods in System Design*. 1996. V. 9, N 1. P. 105–131. DOI: 10.1007/BF00625970.
26. Babar J. et al. GrcatSPN Enhanced with Decision Diagram Data Structures // *Applications and Theory of Petri Nets* / ed. by J. Lilius, W. Penczek. Springer Berlin Heidelberg, 2010. P. 308–317.
27. Chaki S., Gurfinkel A. BDD-Based Symbolic Model Checking // *Handbook of Model Checking* / ed. by E. Clarke et al. Springer, 2018. P. 219–245. DOI: 10.1007/978-3-319-10575-8_8.
28. R. R., S. B., Zimmermann A. An Evaluation Framework for Comparative Analysis of Generalized Stochastic Petri Net Simulation Techniques // *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2020. V. 50. P. 2834–2844. DOI: 10.1109/TSMC.2018.2837643.
29. Pernice S. et al. Multiple Sclerosis Disease: A Computational Approach for Investigating Its Drug Interactions // *Computational Intelligence Methods for Bioinformatics and Biostatistics* / ed. by P. Cazzaniga et al. Cham : Springer International Publishing, 2020. P. 299–308. DOI: 10.1007/978-3-030-63061-4_26.
30. Amparore E., Donatelli S., Landini E. Modelling and Evaluation of a Control Room Application // *Application and Theory of Petri Nets and Concurrency* / ed. by W. van der Aalst, E. Best. Cham: Springer International Publishing, 2017. P. 243–263.
31. Richard L. Performance Results for the CSMA/CD Protocol Using GrcatSPN // *Journal of Systems and Software*. 1997. V. 37, N 1. P. 75–90. DOI: 10.1016/S0164-1212(96)00041-6.
32. Amparore E., Donatelli S. GreatTeach: A Tool for Teaching (Stochastic) Petri Nets // *Application and Theory of Petri Nets and Concurrency*. Springer International Publishing, 2018. P. 416–425. DOI: 10.1007/978-3-319-91268-4_24.
33. Castagno P. et al. A computational framework for modeling and studying pertussis epidemiology and vaccination // *BMC Bioinformatics*. 2020. V. 21, N 8. P. 344. DOI: 10.1186/S12859-020-03648-6.
34. The GrcatSPN Framework. [El. Res.]: <https://github.com/greatspn/SOURCES>. Accessed: 2024-03-15.
35. Paolieri M. et al. The ORIS Tool: Quantitative Evaluation of Non-Markovian Systems // *IEEE Trans. Software Eng.* 2021. V. 47, N 6. P. 1211–1225. DOI: 10.1109/TSE.2019.2917202.
36. Carnevali L., Paolieri M., Vicario E. The ORIS tool: app, library, and toolkit for quantitative evaluation of non-Markovian systems // *ACM SIGMETRICS Performance Evaluation Review*. 2022. V. 49, N 4. P. 81–86. DOI: 10.1145/3543146.3543164.
37. Stewart W. Introduction to the Numerical Solution of Markov Chains. Princeton University Press, 1995. DOI: 10.1515/9780691223384.
38. Carnevali L. et al. Non-Markovian Performability Evaluation of ERTMS/ETCS Level 3 // *Computer Performance Engineering - 12th European Workshop, EPEW 2015*. V. 9272 / ed. by M. Beltran, W. Knottenbelt, J. Bradley. Cham : Springer, 2015. P. 47–62. (Lecture Notes in Computer Science), DOI: 10.1007/978-3-319-23267-6_4.
39. Biagi M. et al. Model-Based Quantitative Evaluation of Repair Procedures in Gas Distribution Networks // *ACM Trans. Cyber Phys. Syst.* 2019. V. 3, N 2. P. 19:1–19:26. DOI: 10.1145/3284037.
40. Carnevali L., Tarani F., Vicario F. Performability Evaluation of Water Distribution Systems During Maintenance Procedures // *IEEE Trans. Syst. Man Cybern. Syst.* 2020. V. 50, N 5. P. 1704–1720. DOI: 10.1109/TSMC.2017.2783188.
41. Carnevali L. et al. Using the ORIS Tool and the SIRIO Library for Model- Driven Engineering of Quantitative Analytics // *Computer Performance Engineering* / ed. by K. Gilly, N. Thomas. Cham: Springer International Publishing, 2023. P. 200–215. DOI: 10.1007/978-3-031-25049-1_13.
42. ORIS Tool. [El. Res.]: <http://www.oris-tool.org>. Accessed: 2024-03-15.
43. ORIS Tool: The Sirio Library. [El. Res.]: <https://github.com/oris-tool/sirio>. Accessed: 2024-03-15.

44. Heiner M. et al. Snoopy — A Unifying Petri Net Tool // Application and Theory of Petri Nets. PETRI NETS 2012. V. 7347 / ed. by S. Haddad, L. Pomello. Springer Berlin Heidelberg, 2012. P. 398–407. (Lecture Notes in Computer Science), DOI: 10.1007/978-3-642-31131-4_22.
45. David R., Alla H. Discrete, Continuous, and Hybrid Petri Nets. Springer Berlin Heidelberg, 2010. P. 550. DOI: 10.1007/978-3-642-10669-9.
46. Liu F., Heiner M., Gilbert D. Fuzzy Petri nets for modelling of uncertain biological systems // Briefings in Bioinformatics. 2018. Dec. V. 21, N 1. P. 198–210. DOI: 10.1093/bib/bbyl18.
47. Fujita M., McGeer P., Yang J. Multi-Terminal Binary Decision Diagrams: An Efficient DataStructure for Matrix Representation // Form. Methods Syst. Des. USA, 1997. Apr. V. 10, N 2/3. P. 149–169. DOI: 10.1023/A: 1008647823331.
48. Hucka M. et al. Systems Biology Markup Language (SBML) Level 2 Version 5: Structures and Facilities for Model Definitions // Journal of Integrative Bioinformatics. 2015. V. 12. N 2. P. 731–901. DOI: 10.2390/biecoll-jib-2015-271.
49. Heiner M., Schwarick M., Wegener J.-T. Charlie — An Extensible Petri Net Analysis Tool // Application and Theory of Petri Nets and Concurrency / ed. by R. Devillers, A. Valmari. Cham : Springer International Publishing, 2015. P. 200–211. DOI: 10.1007/978-3-319-19488-2_10.
50. Heiner M., Rohr C., Schwarick M. MARGIE — Model Checking and Reachability Analysis Done Efficiently // Application and Theory of Petri Nets and Concurrency / ed. by J.-M. Colom, J. Desel. Springer Berlin Heidelberg, 2013. P. 389–399. DOI: 10.1007/978-3-642-38697-8_21.
51. Baier C. et al. Model Checking Continuous-Time Markov Chains by Transient Analysis // Computer Aided Verification / ed. by E. Emerson, A. Sistla. Springer Berlin Heidelberg, 2000. P. 358–372.
52. Donaldson R., Gilbert D. A Model Checking Approach to the Parameter Estimation of Biochemical Pathways // Computational Methods in Systems Biology / ed. by M. Heiner, A. M. Uhrmacher. Springer Berlin Heidelberg, 2008. P. 269–287.
53. Chodak J., Heiner M. Spike — Reproducible Simulation Experiments with Configuration File Branching // Computational Methods in Systems Biology. Springer International Publishing, 2019. P. 315–321. DOI: 10.1007/978-3-030-31304-3_19.
54. Gilbert D., Donaldson R. A Monte Carlo model checker for probabilistic LTL with numerical constraints : tech. reP. / Bioinformatics Research Centre, University of Glasgow. 01/2008.
55. Gilbert D. et al. Spatial quorum sensing modelling using coloured hybrid Petri nets and simulative model checking // BMC Bioinformatics. 2019. V. 20, supplement 4. DOI: 10.1186/sl2859-019-2690-z.
56. Hecrajy M. et al. Snoopy’s hybrid simulator: a tool to construct and simulate hybrid biological models // BMC Systems Biology. 2017. July. V. 11, N 1. DOI: 10.1186/S12918-017-0449-6.
57. Zimmermann A. Modelling and Performance Evaluation with TimeNET 4.4 // Quantitative Evaluation of Systems - 14th International Conference, QEST 2017. V. 10503 / ed. by N. Bertrand, L. Bortolussi. Springer, 2017. P. 300–303. (Lecture Notes in Computer Science), DOI: 10.1007/978-3-319-66335-7_19.
58. Selic B. Modeling And Analysis Of Realtime And Embedded Systems With Uml And Marte Developing Cyberphysical Systems. Elsevier Science & Technology, 2014. DOI: 10.1016/C2012-0-13536-5.
59. Zimmermann A. et al. Analysis of Safety-Critical Cloud Architectures with MultiTrajectory Simulation // Annual Reliability and Maintainability Symposium (RAMS). 01/2022. P. 1–7. DOI: 10.1109/RAMS51457.2022.9893923.
60. Fedorova A., Beliautso V., Zimmermann A. Colored Petri Net Modelling and Evaluation of Drone Inspection Methods for Distribution Networks // Sensors. 2022. V. 22, N 9. DOI: 10.3390/s22093418.

61. Dingle N., Knottenbelt W., Suto T. PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets // SIGMETRICS Performance Evaluation Review ACM. New York, NY, USA, 2009. Mar. V. 36, N 4. P. 34–39. DOI: 10.1145/1530873.1530881.
62. Platform Independent Petri Net Editor v4. [El. Res.]: <https://sourceforge.net/projects/pipe2>. Accessed: 2024-03-15.
63. PIPE 5. [El. Res.]: <https://github.com/sarahtattersall/PIPE>. Accessed: 2024-03-15.

ИНСТРУМЕНТАЛЬНЫЕ СИСТЕМЫ, ПОДДЕРЖИВАЮЩИЕ СТОХАСТИЧЕСКИЕ СЕТЕВЫЕ МОДЕЛИ

А. В. Быстров, И. Б. Вирбицкайте, Е. С. Ошевская
Институт систем информатики им. А. П. Ершова,
630090, Новосибирск, Россия

УДК 004.94+519.876.5

DOI: 10.24412/2073-0667-2024-2-32-57

EDN: KNBRYZ

Стохастические сети Петри — мощное средство моделирования параллельных недетерминированных систем с вероятностными характеристиками, применяемое в самых разных областях человеческой деятельности. Они сочетают наглядность графического представления с развитым математическим и алгоритмическим аппаратом анализа, позволяют изучать не только качественные, но и количественные свойства систем, такие как пропускная способность, надежность, время ожидания и т. п. Разработаны и продолжают появляться новые программные инструменты, поддерживающие создание, модификацию и анализ свойств моделей систем на основе различных вариантов стохастических сетей Петри. В данной работе рассматриваются несколько таких инструментов, доступных в сети Интернет и получивших признание пользователей, обсуждаются предоставляемые ими возможности и проводится их сравнение. Основная цель работы — облегчить исследователю и инженеру выбор наиболее подходящего инструмента моделирования и анализа для решения стоящей перед ним задачи.

Ключевые слова: стохастические сети Петри, моделирование, анализ производительности, инструментальные системы.

Введение. Сети Петри (СП) [1] — одна из наиболее популярных моделей, используемых как для теоретических исследований структуры и поведения параллельных недетерминированных систем, так и для практических применений в различных областях автоматизированной обработки информации: распределенных базах данных и операционных системах, архитектурах вычислительных машин, систем и сетей, системах программного обеспечения, протоколах коммуникаций, системах с элементами искусственного интеллекта, производственных системах, системах экологии и микробиологии, сельскохозяйственном производстве и др. Структура СП состоит из *переходов*, моделирующих действия системы; *мест*, представляющих память действий; *направленных дуг* от входных мест к переходам и от переходов к выходным местам. *Разметка* СП, моделирующая состояние поведения СП, присваивает каждому месту натуральное количество *фишек*. Переход *разрешен* (*к срабатыванию*) *при разметке*, если каждое его входное место содержит достаточное количество фишек; достаточность определяется числом дуг из входного места в переход. При *срабатывании* переход удаляет из каждого входного места столько фишек, сколько дуг идет из места в переход, и добавляет в каждое выходное место столько фишек, сколько дуг идет в место из перехода. Таким образом, срабатывания переходов приводят к смене разметок. При функционировании СП порождается *последовательность срабатываний переходов*. К достоинствам СП относятся наглядное графическое представление

их структуры и эффективные методы анализа их поведения. Однако сети Петри не могут быть использованы для изучения количественных (например, временных) параметров функционирования моделируемых систем, таких как время отклика системы или ее производительность.

Особое место среди моделей с реальным временем занимают *временные сети Петри* (ВСП). В теории сетей Петри известны разнообразные временные расширения, в которых временные характеристики сопоставляются различным элементам (местам, переходам, дугам, фишками) сетевых моделей [2]. Наибольшую популярность получили расширения сетей Петри, где с переходами связываются временные задержки, а модельные часы измеряют ход времени в состояниях сети. В литературе рассматриваются два подхода — *дискретно-временные* СП (ДВСП) и *непрерывно-временные* СП (НВСП). В ДВСП имеются глобальные часы и каждому переходу сопоставляется целочисленное значение — длительность срабатывания перехода, тогда как в НВСП с каждым переходом связываются локальные часы и интервал действительных чисел — временных задержек срабатывания перехода (при этом срабатывание перехода считается мгновенным). Известно, что НВСП равносильны машинам Тьюринга и включают класс ДВСП, поэтому именно модели НВСП вызывают наибольший интерес у исследователей. НВСП имеют следующее поведение. *Состояние* НВСП содержит текущую разметку (распределение фишек по местам сети) и временной вектор показаний часов разрешенных переходов (т.е. переходов, все входные места которых имеют достаточное количество фишек при данной разметке). Переход *разрешен (к срабатыванию)* в состоянии только в том случае, если переход разрешен при этой разметке и его часы достигли момента времени, который находится в пределах временного интервала данного перехода. При функционировании НВСП порождаются *последовательности срабатываний переходов с временными характеристиками*, приводящие к достижимым состояниям. Пространство состояний НВСП, представляемое графом достижимых состояний (графом достижимости), бесконечно и недискретно, что увеличивает сложность анализа модели, поэтому используется метод классов состояний [3]. При функционировании НВСП последовательность срабатываний переходов может запускаться с различными временными характеристиками и при этом достигать одной и той же разметки, но, возможно, с разными временными векторами, содержащими показания часов разрешенных переходов. Множество временных векторов со значениями, попадающими в интервалы разрешенных переходов, принимает форму выпуклого линейного многогранника, обычно называемого *зоной*. *Класс состояний* — это композиция разметки и зоны. *Граф классов состояний (абстракция графа достижимости)* представляет собой символическое представление пространства состояний НВСП, где все часы разрешенных переходов продвигаются с одинаковой скоростью. ВСП пригодны для спецификации и анализа систем реального времени, таких как управление процессами, производственные системы, роботы, авиационные системы.

Другое временное расширение СП — *стохастические сети Петри* (ССП) [4], основанные на концепции стохастических временных задержек. В СП предполагается, что все переходы запускаются с временными задержками — случайными величинами с известными законами распределения. Часто в СП используется экспоненциальный закон распределения времени срабатывания переходов, что упрощает анализ таких СП. В СП с переходами, имеющими неэкспоненциальное распределение задержек, как правило, накладываются существенные ограничения на структуру сетей с целью облегчения их анализа. В настоящее время известно несколько подходов к определению СП. Например,

разработаны ССП с непрерывным временем (временная шкала — неотрицательные действительные числа) [5], обозначаемые НВССП, и дискретным временем (временная шкала — натуральные числа) [6], обозначаемые ДВССП. В *обобщенных ССП* (ОССП) [7] имеются экспоненциальные (с экспоненциально распределенными задержками) и мгновенные (с нулевыми задержками) переходы. *Детерминированные ССП* (ДССП) [8] — это модель с экспоненциальными и детерминированными (с фиксированной задержкой) переходами. *Детерминированные обобщенные ССП* (ДОССП) имеют экспоненциальные, мгновенные и детерминированные переходы. В *расширенных ССП* (РССП) [9–10] возможны переходы с неэкспоненциальными задержками. *Расширенные ДССП* (РДССП) [11] могут иметь детерминированные переходы с неэкспоненциальными задержками. Использование случайных временных величин обогащает анализ систем путем вычисления количественных показателей их эффективности (производительности) и надежности, что позволяет на ранней стадии оценить выбор метода проектирования и предоставляет рекомендации для этапов внедрения и интеграции проектируемых систем. ССП лучше подходят в качестве моделей систем с разделением ресурсов, архитектур «клиент-сервер», коммуникационных систем, производственных линий и т. д. В стохастических сетевых моделях нередко присутствуют так называемые «вознаграждения» (rewards) — функции, которые сопоставляют численные значения состояниям или переходам и которые используются для получения таких характеристик моделируемой системы, как потребление ресурсов, доступность, надежность и др. «Вознаграждения» могут также суммироваться в ходе функционирования модели.

Детальное изложение математического аппарата, примеры и обсуждение различных классов стохастических сетей Петри читатель может найти в работе [12]. Мы также используем аббревиатуры классов, введенные в этой работе.

Анализируют ССП обычно с помощью *марковских процессов* (МП), называемых *марковскими цепями* (МЦ) с дискретным временем (ДВМЦ) для ДВССП и с непрерывным временем (НВМЦ) для НВССП. Граф достижимости ССП может быть непосредственно сопоставлен с МП, который удовлетворяет Марковскому свойству (говоря нестрого, при фиксированном настоящем будущее независимо от прошлого). Каждое состояние в графе достижимости сопоставляется с состоянием в марковском процессе, а срабатывание перехода с некоторой временной задержкой — с переходом в марковское состояние с соответствующей вероятностью. МЦ имеют множество применений в качестве статистических моделей реальных процессов, например, при изучении систем круиз-контроля в автомобилях, очередей клиентов, курсов обмена валют, динамики популяции животных и т. д. Для анализа НВМЦ, как правило, используют численные подходы. Когда в расширенных ССП (РССП) имеются переходы с неэкспоненциальными задержками, лежащий в основе модели стохастический процесс относится к более сложным классам — немарковским процессам [13], но численное решение все еще может быть получено, если модель удовлетворяет Марковскому свойству в определенные моменты времени, называемые *точками регенерации*. В частности, если новая регенерация всегда достигается с вероятностью 1, то модель поддерживает марковский регенерирующий процесс (МРП), который может быть решен численно. Другими словами, МРП — это стохастический процесс, который рано или поздно обязательно достигнет регенерации.

Стохастический процесс имеет как стационарное (устойчивое) состояние, так и переходное. По истечении определенного времени в процессе устанавливается *стационарное состояние*, т. е. когда текущее поведение системы сохранится и в будущем. *Переходное состояние* — это, по сути, время между началом некоторого события (например, сраба-

тиванием переходов) и стационарным состоянием. *Стационарный анализ* стохастического процесса используется для определения стационарных состояний. При *переходном анализе* процесса осуществляется исследование вероятностей нахождения в определенном состоянии в определенный момент времени. Переходный анализ процессов занимает значительно больше времени и требует больших вычислительных мощностей для получения результатов. Регенеративный переходный/стационарный анализ [14–15] — это переходный/стационарный анализ, выполненный в рамках МРП. Для анализа стохастических процессов часто используется симуляция их поведения. При *стационарной симуляции* процессов вычисляется решение, которое не меняется со временем, тогда как при *переходной симуляции* — мгновенные значения в каждый момент времени для каждой случайной величины. Стационарная симуляция происходит намного быстрее, чем переходная. Кроме того, для анализа ССП применяется метод *классов стохастических состояний* [5, 16], объединяющий проверку корректности возможных вариантов поведения (идентифицируемых базовой НВСП) с количественной оценкой их вероятности (индуцируемой стохастическими параметрами НВССП).

Очевидно, что при разработке прикладных проектов необходимо использовать программный инструментарий. Инструменты, поддерживающие ССП, можно использовать для анализа производительности стохастических систем, а также для генерации кода, симуляции, анализа и верификации разных классов ССП. На сегодняшний день существует ряд таких инструментов, разработанных для различных целей и сред (платформ). Однако имеет смысл рассматривать только те инструментальные системы, которые обладают следующими свойствами:

- поддерживают стохастические модели сетей Петри и их обобщения;
- включают графический редактор моделей, симулятор, аниматор, анализатор и верификатор;
- требуют разумные аппаратные и общедоступные программные ресурсы;
- поддерживают мультиплатформенность;
- достаточно хорошо документированы;
- бесплатны, по крайней мере, для академических исследований.

В данной работе дается детальный обзор пяти выдержавших проверку временем, свободно распространяемых и ориентированных на различные платформы инструментальных систем (а именно, GreatSPN, ORIS, PetriNuts, TimeNet, PIPE2), поддерживающих широкий спектр разновидностей стохастических моделей и обеспечивающих их оценку производительности с помощью эффективных методов решения, включая симуляцию стохастических процессов. Рассматриваемые инструменты сравниваются с точки зрения их возможностей функционального и временного анализа моделей, а также обсуждаются рекомендации пользователям по использованию инструментов в зависимости от того, какой тип стохастических моделей требуется исследовать и с какими операционными системами пользователи работают.

1. GreatSPN. GreatSPN [17–18] (изначально разрабатывался в университете г. Торонто (Италия), в настоящее время также поддерживается в университете г. Алессандрии (Италия)) представляет собой многофункциональный инструментальный программный комплекс на базе различных моделей стохастических сетей Петри, предназначенный для спецификации, анализа, валидации и оценки производительности параллельных/распределенных систем, функционирующих в режиме реального времени.

В GreatSPN версии 3.0 поддерживаются следующие стохастические модели:

— обобщенные стохастические сети Петри (ОССП), базовые сети которых включают приоритеты переходов и ингибиторные дуги и которые являются удобным формализмом для оценки производительности больших систем благодаря возможности их автоматического преобразования в непрерывно-временную Марковскую цепь (НВМЦ);

— ОССП с цветными фишками (ОЦССП) [19], которые расширяют возможности моделирования многокомпонентных систем;

— ОССП с детерминированными переходами (ДОССП) и с неэкспоненциальными переходами (РОССП).

Модели могут быть модульными, т. е. состоять из отдельных независимых сетей, которые объединяются с помощью алгебраических операторов композиции. Цель анализа поддерживаемых стохастических моделей состоит в том, чтобы проверить их правильность с помощью структурного/поведенческого анализа и проверки модели (model checking). Как только пользователь уверен, что модель отражает поведение проектируемой системы, процесс анализа концентрируется на вероятностных аспектах посредством стохастической проверки модели и вычисления классических показателей оценки производительности и надежности.

Функциональные возможности системы GreatSPN включают: структурный анализ, исследование пространства состояний, преобразования между поддерживаемыми формализмами, верификацию, поддержку принятия решения на основе Марковских цепей, численные решения, симуляцию методом Монте-Карло.

Система GreatSPN представляет собой набор инструментов, которые могут быть организованы по их функциям в следующие категории:

- унифицированный графический интерфейс, реализованный на Java;
- инструменты трансляции, композиции, развертки моделей для работы с несколькими форматами;
- структурный анализатор свойств моделей;
- генератор и оптимизатор Марковских процессов (цепей) для анализа стохастических моделей;
- проверка моделей (model checker) на базе темпоральных логик;
- стохастический решатель.

Рассмотрим подробнее возможности и особенности каждого инструмента.

Графический редактор является центральным модулем системы GreatSPN, поскольку он используется для построения моделей в графическом представлении и определения их свойств, а также для интерактивной симуляции с анимацией «игры с (цветными) фишками» модели. Он отвечает также за вызов командной строкой всех основных модулей (моделирования, верификации, анализа, симуляции и т. д.) и за визуализацию результатов их работы. С целью более удобного обмена моделями с другими инструментами система GreatSPN поддерживает стандартный формат PNML [20–21]. К сожалению, в действующей версии PNML еще нет надлежащего способа представления стохастических атрибутов элементов сети, и при экспорте модели в формате PNML такие атрибуты опускаются. Основным форматом ввода остается нестандартный, но широко поддерживаемый формат net/def, при этом доступны и другие форматы (APNN, GrML, NetLogo). Возможна трансляция из PNML в net/def и из UML в сети Петри, развертка ОЦССП в ОССП, а также композиция сетевых моделей.

Структурный анализатор включает модули:

- вычисления Р/Т-инвариантов/потоков;

- вычисления структурной ограниченности и неограниченных последовательностей переходов;
- вычисления структурных конфликтных множеств, тупиков, ловушек и взаимоблокировок;
- анализ сетевых свойств, представленных формулами темпоральных логик LTL , CTL , CTL^* [22].

Модуль проверки моделей (model checker) включает модуль символьной проверки моделей (symbolic model checker) [23] на базе темпоральной логики CTL^* и модуль стохастической проверки моделей (stochastic model checker) на базе непрерывно-временной стохастической темпоральной логики CSL^{TA} для временных автоматов, в которой вычисленные показатели производительности выражаются с использованием детерминированных временных автоматов. Возможно графическое представление таких автоматов в интерфейсе системы GreatSPN, а затем их использование для вычисления показателей производительности моделей.

Генератор и оптимизатор Марковских процессов (цепей) осуществляет их построение из высокоуровневого языка Markov Decision Petri Nets [24], предназначенного для спецификации вероятностного и недетерминированного поведения моделируемой системы и синтаксического способа определения переключения между этими двумя типами поведения, а также вычисляет решение Марковских процессов с использованием модулей как переходного, так и стационарного численного анализа для получения показателей производительности из Марковских цепей.

Стохастический решатель включает в себя набор инструментов для проведения анализа с использованием графов достижимости, симуляции и дифференциальных уравнений.

Численные решения для переходных и стационарных процессов используют построение графов достижимости сетевых моделей. Существует несколько разновидностей графов достижимости, которые можно строить в системе:

- граф достижимости ОССП-моделей;
- граф достижимости ОЦССП-моделей;
- граф достижимости символьных разметок (объединяющих разметки, при которых разрешены одни и те же переходы) в ОЦССП, использующий симметрии моделей [25];
- граф достижимости марковских регенерирующих процессов (МРП) стохастических моделей с возможностью одновременных срабатываний переходов;
- граф достижимости, закодированный с применением диаграмм принятия решений (Binary Decision Diagrams) [26–27].

Решатель для ДОССП использует расширенный формализм сетевой модели, который включает параметры разметки, параметры скорости переходов, особенности срабатывания переходов (например, задержка перехода может определяться функцией, зависящей от разметки), общее, возможно неэкспоненциальное, время задержки срабатывания переходов и предохранители переходов (guards). Этот решатель может обрабатывать как НВМЦ в переходном и стационарном режимах, так и МРП только в стационарном режиме и использует передовые численные методы.

Симулятор Монте-Карло позволяет вычислять показатели производительности с использованием пакетной симуляции в переходном или стационарном режиме. Для ОЦССП симуляция может быть выполнена либо с использованием обычных разметок и переходов, либо с использованием символьных разметок. Симулятор поддерживает стохастиче-

ские модели с произвольным числом переходов с различными распределениями задержек срабатывания переходов (равномерное, Эрланга, Кокса и др.). Сравнительный анализ возможностей симулятора GreatSPN можно найти в статье [28], где показана высокая точность и производительность алгоритмов анализа, используемых симулятором. Инструмент ориентируется на вычисление целевых показателей производительности и выполняет итерации по созданным событиям до тех пор, пока все показатели не окажутся ниже порогового значения точности.

Сгенерированные показатели производительности выражаются с использованием языка, специфичного для предметной области, могут быть экспортированы в форматах CSV, Excel, PDF, PNG для обработки с помощью внешних инструментов.

Предусмотрены два основных типа взаимодействия GreatSPN с другими инструментами:

- экспорт моделей из GreatSPN в принимающий инструмент, чтобы применять методы анализа, доступные в этом инструменте;

- повторное использование модулей GreatSPN (или их части) другими инструментами.

Инфраструктура GreatSPN3.0 использовалась в таких контекстах как биологическое моделирование [29], проектирование диспетчерского центра в газовой компании [30], оценивание эффективности сетевых протоколов [31] и многих других. В ходе разработки GreatSPN особое внимание уделялось удобству использования этого инструмента при обучении работе со стохастическими моделями [32] и для изучения биологических систем [33].

Дистрибутив системы GreatSPN включает также дополнительные утилиты:

- *algebra* предназначена для композиции моделей и визуализации графов достижимости;

- *multisolve* предназначена для серийного запуска анализатора с варьируемыми параметрами модели.

GreatSPN имеет открытый исходный код, доступный [34] на GitHub, и работает на всех основных платформах (Linux, Windows, macOS).

2. ORIS. ORIS [35–36] (разрабатывался и поддерживается Университетом г. Флоренции (Италия) и Университетом Южной Калифорнии г. Лос Анжелеса (США)) — это программный инструмент для моделирования и анализа производительности и надежности стохастических систем, управляемых немарковскими таймерами (например, временем между прибытием и/или обслуживанием, временем отказа, временем ремонта, колеблющимися задержками, тайм-аутами), задаваемыми с использованием различных функций плотности вероятности.

В целом, ORIS представляет собой хорошо спроектированный программный инструмент, полностью реализованный на Java и разработанный таким образом, чтобы облегчить его расширение новыми средствами работы с непрерывно-временными стохастическими сетями Петри (НВССП) и реализацию новых методов решения, лежащих в их основе стохастических процессов.

В качестве графического формализма для моделирования стохастических систем ORIS использует НВССП, в базовых НВСП которых с каждым переходом могут быть связаны:

- функция, контролирующая количество фишек во входных местах, необходимое для срабатывания перехода;

- функция, определяющая дополнительное количество фишек в выходных местах после срабатывания перехода;

- множество переходов, для которых принудительно переустанавливается время с целью их срабатывания;
- приоритет и вес (при конкуренции на срабатывания выбираются переходы с наивысшим приоритетом и с вероятностью, пропорциональной их весам);
- временные интервалы срабатывания переходов (т.е. возможно задание как непрерывно-временных задержек, так и нулевых).

В НВССП функции плотности вероятности, связанные с переходами, могут быть немедленными, детерминированными, экспоненциальными и неэкспоненциальными. Следует подчеркнуть, что ORIS поддерживает НВССП, в каждом состоянии которых несколько таймеров могут быть запущены одновременно.

ORIS предоставляет как графический пользовательский интерфейс (GUI), так и интерфейс прикладного программирования (Java API), основанный на встроенной Java-библиотеке SIRIO.

Графический пользовательский интерфейс предназначен для моделирования стохастических систем в виде НВССП с помощью графического редактора, симуляции НВССП (с возможной анимацией) и количественной оценки НВССП, выполняемой с использованием разнообразных механизмов принятия решений, применимых при различных предположениях о лежащих в основе моделей стохастических процессов. При этом Java-библиотека SIRIO позволяет интегрировать методы анализа, доступные в графическом интерфейсе, в пользовательские программы и инструменты для выполнения обширных параметрических исследований, где свойства систем (такие как производительность, надежность, пригодность) оцениваются при комбинациях многих параметров и вариантов моделей, специфицированных с помощью Java API.

Чтобы упростить такую интеграцию, модели НВССП можно экспортировать из графического интерфейса в виде Java-кода. Как только модель построена с использованием Java API, ее можно анализировать, вызвав доступные анализаторы, и результаты (например, стационарные распределения, временные ряды переходных процессов, графы классов состояний) могут быть сохранены в файл или показаны с использованием функций, доступных в графическом редакторе. В то время как построение и анализ моделей непосредственно в графическом редакторе полезны для быстрого перебора различных вариантов моделей и анализа их поведения, Java API дает возможность, варьируя стохастические параметры, проводить обширные параметрические исследования моделей.

ORIS предоставляет пользователю широкий выбор возможностей стохастического анализа, которые реализуют различные методы решения и оценки свойств НВССП, заданных с использованием одного и того же формата ввода параметров (например, «вознаграждений»¹ и условий остановки²). ORIS также поддерживает анализ НВСП, основанный на абстракции поведения НВСП — графе классов состояний. Каждый метод анализа накладывает различные ограничения на класс лежащих в основе НВССП стохастических процессов, что приводит к более эффективным методам решения.

Рассмотрим методы стохастического анализа, доступные в ORIS.

¹«Вознаграждения» — функции, которые сопоставляют численные значения состояниям или переходам и которые используются для получения таких характеристик моделируемой системы, как потребление ресурсов, доступность, надежность и др.

²Условие остановки — логическое выражение, при истинности которого анализатор останавливает анализ до истечения ранее назначенного срока.

Марковский метод реализует стандартные методы переходного и стационарного анализа непрерывно-временных цепей Маркова (НВЦМ) [37]. Возможен анализ только ОССП, т. е. допустимы только экспоненциальные и мгновенные переходы; на практике, чтобы преодолеть это ограничение, каждый неэкспоненциальный переход (т. е. переход с временной задержкой, заданной неэкспоненциальным распределением) может быть заменен (перед анализом) последовательностью экспоненциальных переходов, полученных с помощью аппроксимации фазового типа.

Метод ограничения готовности к срабатыванию переходов реализует переходный анализ для марковских регенерирующих процессов (МРП). Возможен анализ только НВССП, в каждом состоянии которых не более чем один неэкспоненциальный переход разрешен к срабатыванию.

Регенеративный метод реализует переходный и стационарный анализ МРП с несколькими неэкспоненциальными переходами, разрешенными к срабатыванию в состояниях. Для стационарного анализа требуется ограниченное число срабатываний переходов до тех пор, пока не произойдет регенерация — состояние, в котором все переходы будут вновь готовы к срабатыванию, тогда как для переходного анализа этого ограничения нет.

«*Направленный*» метод реализует переходный анализ НВССП без ограничений на наличие регенераций и при этом перебирает классы состояний в графе классов стохастических состояний, начиная с начального состояния, до тех пор, пока самое раннее время срабатывания переходов в каждой последовательности срабатываний переходов не превысит целевую временную границу анализа.

Недетерминированный метод анализа пространства состояний НВССП поддерживает проверку качественных свойств модели: например, проверку того, может ли быть достигнута конкретная разметка или содержит ли граф классов состояний регенерацию в каждом цикле (это позволяет проводить более точный регенеративный анализ).

На протяжении многих лет ORIS успешно используется в различных контекстах и областях применения, например, в качестве графического интерфейса (GUI) для оценки показателей работоспособности в системах железнодорожной сигнализации [38], в качестве интерфейса прикладного программирования для выполнения анализа технического обслуживания в газо- и водо-распределительных сетях [39–40], для оценки наполняемости вагонов и времени ожидания пассажиров трамвайной линии [41].

Инструмент ORIS находится в свободном доступе [42], а исходный код библиотеки SIRIO доступен по лицензии AGPL на GitHub [43].

3. PetriNuts. PetriNuts (разработан и продолжает развиваться в Бранденбургском техническом университете г. Котбус, Германия) — программный комплекс, включающий инструменты: Snoory, Charlie, Marcie, Spike, MC2, которые поддерживают работу с различными классами моделей сетей Петри и их стохастическими расширениями.

Snoory [44] — базовый инструмент комплекса, предназначенный для графического конструирования, редактирования и визуализации, а также симуляции (с возможной анимацией «игры с фишками») сетевых моделей. Инструмент поддерживает следующие классы моделей:

- обычные сети Петри (СП);
- сети Петри с ингибиторными (inhibitor)/читающими (read)/обнуляющими (reset)/зависящими от маркировки дугами;
- непрерывные (continuous) сети Петри (НСП) [45];
- дискретно- и непрерывно-временные сети Петри (ДВСП и НВСП);

- стохастические непрерывные сети Петри (СНСП);
- обобщенные стохастические дискретно-временные сети Петри (ОДВССП);
- нечеткие (fuzzy) расширения НСП и СНСП [46];
- вышеперечисленные сетевые модели с цветными фишками.

Помимо сетевых моделей названных классов в графическом интерфейсе Snoору также можно строить и визуализировать и другие графы, в частности, графы достижимости, мульти-терминальные и интервальные бинарные диаграммы решений [47], деревья неисправностей.

Для обмена данными с другими инструментами комплекса и сторонними средствами анализа Snoору предоставляет несколько форматов файлов представления моделей, в частности, стандартные форматы PNML и SBML (Systems Biology Mark-up Language) [48].

Charlie [49] — реализованный на Java инструмент анализа сетевых моделей. Он принимает на вход все поддерживаемые в Snoору классы моделей, но временные и стохастические атрибуты игнорируются, рассматривается только структура моделей. Инструмент предоставляет простой графический интерфейс, в котором можно определять тип сети (конечный автомат, маркированный граф, сеть со свободным выбором и др.), выяснять ее структурные свойства, находить Р/Т инварианты, ловушки и тупики, строить и визуализировать покрывающий граф и граф достижимости. На основе этих графов анализируются такие поведенческие свойства, как ограниченность, живость, обратимость, наличие динамических конфликтов, а также осуществляется проверка моделей на выполнимость формул темпоральных логик *LTL* и *CTL* [22]. Набор анализаторов и форматов ввода анализируемых сетевых моделей может быть расширен с помощью подключаемых модулей.

Marcie [50] — инструмент анализа ССП и ОССП, в которых можно дополнительно задавать «вознаграждения». Инструмент позволяет производить как переходный, так и стационарный анализ стохастических процессов. В зависимости от класса сети и размера пространства состояний для анализа могут применяться различные внутренние вычислители: точный численный (только для ограниченных сетей), аппроксимирующий (при работе которого маловероятные состояния отбрасываются) и симуляционный. В каждом из вычислителей реализовано несколько алгоритмов, так что пользователь может подбирать наиболее подходящий для конкретной модели. Кроме того, с использованием этих вычислителей может выполняться проверка моделей на базе формул темпоральных логик *CTL*, *CSRL* (Continuous Stochastic Reward Logic) [51] и *PLTLc* (Probabilistic Linear-time Temporal Logic with numerical constraints) [52].

Spike [53] — инструмент командной строки для эффективной симуляции стохастических, непрерывных и стохастических непрерывных сетей Петри и их расширений цветными фишками. Инструмент предоставляет разнообразные алгоритмы симуляции, позволяет определять сложные сценарии сеансов симуляции с варьируемыми параметрами и точно воспроизводить сохраненные сценарии.

MC2 [54] — реализованный на Java инструмент проверки моделей, свойства которых заданы формулами темпоральных логик *LTLc* и *PLTLc* на множестве трасс, полученных в результате симуляции стохастических и стохастических непрерывных СП и их расширений цветными фишками. Такие трассы могут быть получены, например, с помощью инструментов Snoору и Spike. Пример использования системы MC2 в моделировании роста биопленок можно найти в [55].

Все инструменты программного комплекса PetriNuts свободно доступны для некоммерческого использования. Особенно активно и успешно этот комплекс применяется для моделирования биологических систем (см., например, [56]).

4. TimeNET. TimeNET [57] (с 90-х годов разрабатывался в Берлинском Техническом Университете, а с 2008 года поддерживается и развивается в Техническом Университете г. Ильменау (Германия)) — это программный инструмент для моделирования и оценки производительности систем с использованием стохастических сетей Петри и цепей Маркова. Основными особенностями инструмента являются: возможность оценки стохастических моделей с неэкспоненциально распределенными задержками срабатывания переходов, работа со стохастическими сетями Петри с цветными фишками и эффективные методы симуляции для моделей с «редкими» (происходящими с низкой вероятностью) событиями.

В TimeNET поддерживаются следующие классы моделей:

- обобщенные стохастические сети Петри (ОССП);
- детерминированные ОССП (ДОССП), расширяющие ОССП переходами с фиксированными задержками, при ограничении, что в любой разметке сети не более чем один такой переход готов к срабатыванию³;
- расширенные детерминированные стохастические сети Петри (РДССП), где допускаются переходы с неэкспоненциальным распределением задержек срабатывания. Многие известные распределения (равномерное, треугольное, усеченное экспоненциальное, конечно-дискретное) относятся к этому классу распределений;
- стохастические сети Петри с цветными фишками (ЦССП);
- диаграммы состояний UML-профиля для MARTE (Modeling and Analysis of Real-Time and Embedded systems) [58], которые транслируются в РДССП для последующего анализа.

TimeNET спроектирована как модульная и расширяемая система, центральной частью которой является графический интерфейс пользователя (модуль GUI), реализованный на Java. Он служит графическим редактором моделей, средством запуска и управления различными модулями анализа и симуляции, написанными на C++ и C, демонстратором результатов их работы, а также интерактивным визуализатором динамического поведения моделей. Модуль GUI является универсальным в том отношении, что поддерживается единообразная работа с различными классами моделей. Каждый класс описывается XML-схемой, которая не только неявно определяет формат файла модели, но задает и графический интерфейс, включая отрисовку элементов модели и иконки, показываемые пользователю. Конкретная модель хранится в виде XML-файла, проверяемого на соответствие схеме. Для взаимодействия остальных модулей системы с модулем GUI разработан специальный интерфейс, через который каждый модуль может сообщать конкретный класс моделей, с которым он работает, и дополнять меню GUI специфичными командами. Такой подход позволяет легко расширять систему, сохраняя унифицированный интерфейс пользователя. Модули анализа и симуляции могут также запускаться командной строкой и скриптами.

Для класса РДССП, включающего ССП и ОССП как подклассы, поддерживаются следующие возможности:

- автоматическая/интерактивная симуляция в модуле GUI;

³Ограничение применяется лишь к численным методам анализа, но не к симуляции.

- структурный анализ (нахождение Р/Т инвариантов, ловушек, тупиков, конфликтных множеств, оценка размера пространства состояний);
- визуализация графа достижимости;
- численный анализ стационарного поведения;
- численный анализ переходного поведения;
- симуляция стационарного поведения:
- с автоматическим определением начала стационарной фазы и остановкой при достижении заданной точности;
- со специальными методами для моделей с «редкими» событиями;
- с комбинированием с численным анализом;
- симуляция переходного поведения;
- построение сложных показателей производительности, включающих вероятности количества фишек в местах сети и срабатываний переходов;
- автоматический обсчет параметризованной модели с варьируемым параметром;
- импорт/экспорт модели в стандартном формате PNML.

Для класса ЦССП поддерживаются следующие возможности:

- автоматическая/интерактивная симуляция в модуле GUI;
- симуляция стационарного поведения с эвристической остановкой по точности и специальными методами для «редких» событий;
- симуляция переходного поведения с контролем точности;
- построение сложных показателей производительности, включающих вероятности количества фишек в местах сети и срабатываний переходов;
- хранение параметров сложных моделей в базе данных.

Система TimeNet применялась для моделирования различных программных, аппаратных и промышленных систем, например, при проектировании архитектуры облачной инфраструктуры управления железнодорожным транспортом [59] и моделировании мониторинга с помощью дронов [60]. Система TimeNET доступна для установки на 32/64-бит Windows и Linux ОС при условии некоммерческого использования.

5. PIPE2. Инструмент PIPE2 (Platform-Independent Petri Net Editor 2) [61], изначально разработанный в Imperial College London, а затем усовершенствованный/адаптированный усилиями других организаций, а также отдельных программистов, реализован на Java для работы с моделями на основе обычных сетей Петри (СП), обобщенных стохастических сетей Петри (ОССП) и их расширений цветными фишками (ОЦСП). Инструмент предоставляет интуитивно ясный графический интерфейс для построения модели, ее интерактивной симуляции, сохранения и загрузки в стандартном формате PNML, а также для вызова модулей, выполняющих различные виды анализа. Модульная структура PIPE2, фиксированный интерфейс взаимодействия GUI с модулями анализа и открытость исходного кода предполагают возможность достаточно простого расширения функциональности инструмента.

С поставляемым набором модулей доступны следующие возможности:

- структурный анализ, включающий:
 - определение типа сетевой модели (конечный автомат, маркированный граф, (расширенная) сеть со свободным выбором и (расширенная) простая сеть);
 - вывод матрицы инцидентности сети Петри;
 - нахождение Р/Т инвариантов, минимальных ловушек и тупиков;
- поведенческий анализ, включающий:

- определение свойств живости, безопасности и наличия тупиковых ситуаций;
- построение и визуализацию покрывающего графа и графа достижимости;
- интерактивная/автоматическая анимация;
- численный анализ стационарного поведения, включающий нахождение:
 - множества существенных состояний (разметок, в которых разрешены только экспоненциальные переходы) и вероятностей этих состояний;
 - времени пребывания в каждом состоянии;
 - среднего количества фишек для каждого места сети;
 - вероятности присутствия конкретного количества фишек для каждого места сети;
 - среднего количества срабатываний за единицу времени для каждого экспоненциального перехода;
- симуляция Монте-Карло для вычисления среднего количества фишек в каждом месте;
- построение с помощью графического языка сложных показателей производительности, которые могут затем вычисляться сторонними средствами анализа.

Следует заметить, что работа поставляемых модулей анализа ОЦСП осуществляется путем игнорирования фишек всех цветов, кроме одного, базового.

Модульность PIPE2, фиксированный интерфейс взаимодействия модуля GUI с модулями анализа и открытость исходного кода дают возможность достаточно простого расширения функциональности инструмента.

Доступны две основные версии PIPE2: v4 [62] и v5 [63]. Первая из них получила широкое распространение, но развивалась довольно хаотично, в результате чего исходный код этой версии сложно понимать и модифицировать. Версия v5 переписана заново, архитектурно более правильно, с избавлением от некоторых ошибок, имеющих в версии v4, однако большая часть модулей анализа в версию v5 не перенесена.

6. Сравнение инструментов. В табл. 1 приведены результаты сравнения рассмотренных выше инструментальных систем на основе пяти критериев.

По первому критерию происходит сравнение с точки зрения того, какие *стохастические модели* поддерживаются инструментом. Все инструменты, за исключением Charlie, позволяют работать с СП и ОСП. В системе ORIS базовой моделью являются НВСП, тогда как в инструменте Snoору предоставляется возможность моделирования ДВСП. Детерминированные переходы могут присутствовать в стохастических моделях систем GreatSPN, ORIS и TimeNet, так как поддерживаются модели ДСП и ДОСП. Кроме того, системы GreatSPN, ORIS и TimeNet наиболее привлекательны при моделировании и анализе производительности и надежности стохастических систем, описываемых немарковскими процессами, поскольку эти инструменты ориентированы на работу с РСП, т. е. пользователь может использовать переходы с неэкспоненциальными распределениями. Почти все инструменты, кроме ORIS, Marcie и Charlie, поддерживают цветные расширения стохастических моделей. Следует подчеркнуть, что модули комплекса PetriNuts позволяют работать со стохастическими вариантами широкого спектра модификаций сетей Петри (СП с ингибиторными/читающими/обнуляющими/зависящими от маркировки дугами, СП с элементами нечетких логик, СП, в которых условия запуска переходов формулируются на основе возможно нецелочисленных параметров).

Вторым критерием сравнения являются (*нестохастические*) *функции* инструментов. Большинство инструментов предоставляют графический редактор, позволяющий строить,

Таблица 1

Сравнение инструментов

		GreatSPN	Oris	PetriNuts					TimeNET	PIPE2
				SNOOPY	MARCIE	CHARLIE	Spike	MC2		
Модели ССП	СП			•	•	•				•
	ССП, ОССП	•	•	•	•		•	•	•	•
	ДВССП			•						
	НВССП		•							
	ДССП, ДОССП	•	•						•	
	РССП, РДССП	•	•						•	
	ЦССП, ОЦССП	•		•			•	•	•	•
	СНСП			•			•	•		
Функциональность	Графический редактор	•	•	•		•			•	•
	Анимация	•	•	•					•	•
	Структурный анализ	•				•			•	•
	Граф достижимости	•				•			•	•
	Абстракция графа достижимости	•	•							
	Проверка моделей	•			•	•		•		
	Поддержка PNML	•		•	•		○		•	•
	Расширяемость	○	○		•	•				○
Стохастика	Численные решатели	•			•				•	•
	Симуляция стохастических процессов	•	•		•				•	○
	Сложные показатели производительности	•	•		•				•	○
Платформы	Java		•			•		•		•
	Linux	•	•	•	•	•	•	•	•	•
	Windows	•	•	•	•	•	•	•	•	•
	Mac	•	•	•	•	•	•	•		•
Доступность		О	С/О	НК	НК	НК	НК	НК	НК	О

редактировать и визуализировать структуру сетевых моделей, и интерактивный/автоматический симулятор сетевого поведения, в котором, как правило, анимация «игры с фишками» может управляться пользователем (выбор срабатывающих из набора разрешенных переходов осуществляется в интерактивном режиме) или может быть делегирована инструменту (случайный выбор). Такие функции инструментов могут помочь обучающимся работе с сетевыми моделями лучше понять их особенности и достоинства/недостатки при анализе моделируемых систем.

Системы GreatSPN, Charlie, Time Net и PIPE2 имеют широкий спектр возможностей структурного анализа сетевых моделей для определения Р/Т-инвариантов/поток, структурной ограниченности, структурных конфликтных множеств, тупиков, ловушек,

взаимоблокировок и т. д. В GreatSPN, в отличие от других инструментов, можно также осуществлять структурный анализ на основе формул темпоральных логик, представляющих сетевые свойства.

В системах GreatSPN, Charlie, TimeNET, PIPE2 строятся графы достижимых состояний моделей, с помощью которых устанавливаются качественные поведенческие свойства: ограниченность, живость, достижимость и т. д. Особенностью GreatSPN является то, что различные типы графов достижимости используются также и для стохастического анализа.

Хорошо известно, что при анализе реальных систем возникает проблема «взрыва пространства состояний», с которой можно бороться несколькими способами, среди которых есть методы, основанные на построении абстракций, эквивалентных графам достижимости относительно некоторых наборов свойств. В GreatSPN, используя симметрию в моделях ОЦССП, строится символьный граф достижимости, каждая вершина которого соответствует набору состояний, в которых разрешены к срабатыванию одни и те же переходы, тогда как в ORIS выполняется анализ НБССП с применением графа класса состояний, в котором каждой вершине соответствует набор состояний, имеющих одну и ту же разметку и схожие временные характеристики.

Две характеристики, которые уникальны для GreatSPN, — это наличие модулей символьной проверки (symbolic model checker) на базе темпоральной логики CTL^* и стохастической проверки (stochastic model checker) на базе непрерывно-временной стохастической темпоральной логики CSL^{TA} . В Charlie возможна проверка только качественных (безвременных) свойств сетевых моделей. С другой стороны, отличительной особенностью Charlie является выявление свойств сети на основе теорем теории сетей Петри, т. е. свойств, которые могут быть выведены из наличия/отсутствия уже выявленных свойств у конкретной сети и которые становятся известными без дополнительного анализа. В инструментах Marcie и MC2 особое внимание уделено анализу стохастических свойств (например, возможно учитывать «вознаграждения» и численные ограничения (constraints)), представленных формулами логик $CSRL$ и $PLTLc$.

К существенным функциям инструмента относятся поддержка стандартного формата PNML, присутствующая в GreatSPN, TimeNet, PIPE2 и частично в PetriNuts (в таблице обозначается символом \circ), а также возможность расширения функциональности инструмента пользователем, которая в Marcie и Charlie реализована в виде поддержки подключаемых модулей (plugins), а в GreatSPN, PIPE2 и Oris — за счет открытости исходного кода, что, впрочем, требует более серьезных усилий. Отметим, что в Oris исходный код открыт только для библиотеки Sirio (обозначается символом \circ).

Третий критерий сравнения — это *возможности стохастического анализа* сетевых моделей, предоставляемые инструментами. Не всегда легко подобрать подходящий решатель и способы его использования при наличии широкого выбора доступных решателей, поэтому мы кратко перечислим характерные особенности таких модулей в рассмотренных инструментах. Численные решатели стохастического поведения реализованы в системах GreatSPN, ORIS, Marcie, TimeNet, PIPE2. Отличительной особенностью ORIS является возможность анализа стохастических моделей, в которых в каждом состоянии может быть параллельно включено несколько таймеров с неэкспоненциальным распределением, а также есть возможность работы с немарковскими процессами. Это значительно расширяет класс моделей, поддающихся численному решению, но требует более высоких вычислительных затрат, и поэтому при работе с инструментом следует выбирать наиболее под-

ходящий для конкретных целей метод анализа. На данный момент ORIS предоставляет самые передовые решатели среди доступных. Следует отметить, что ORIS и Marcie позволяют осуществлять анализ моделей, в которых можно дополнительно задавать «вознаграждения». Возможности включения нескольких таймеров с неэкспоненциальным распределением нет в других инструментах, и только в TimeNet и GreatSPN реализованы методы для оценки моделей, в каждом состоянии которых может быть включен не более чем один такой таймер. Особенность GreatSPN — это эффективные методы численного решения, основанные на симметриях для ОЦСП. TimeNet сосредоточен на численном анализе моделей РДССП в стационарном и переходном режимах. Особенность системы PIPE2 состоит в том, что она очень компактна и проста в работе.

Симуляция стохастических процессов поддерживается в GreatSPN, Marcie, Spike, TimeNet, PIPE2. Симулятор Монте-Карло инструмента GreatSPN вычисляет показатели производительности в переходном/стационарном режиме. Отличительной особенностью инструмента TimeNet являются эффективные методы стохастической симуляции для моделей с «редкими» (происходящими с низкой вероятностью) событиями. Spike предоставляет различные алгоритмы стохастической симуляции с заданием и сохранением сложных симуляционных сценариев, выполняемых с переменными параметрами.

Построение сложных показателей производительности (например, комбинация присутствия конкретного количества фишек в местах сети и среднего количества срабатываний переходов) выполняется в GreatSPN, ORIS, TimeNET и частично в PIPE2 (в таблице обозначается символом o).

Следующий критерий — это *платформы*, поддерживаемые инструментом. Все рассмотренные инструменты либо универсальны (реализованы на Java), либо имеют версии для всех трех основных платформ: Windows, Linux и Mac, за исключением того, что у TimeNET нет версии для Mac.

Последний критерий — это *доступность*, т. е. условия, на которых можно использовать инструмент. Все компоненты PetriNuts и система TimeNET поставляются в бинарном виде и свободно доступны для некоммерческого использования (в таблице обозначается НК). Инструменты GreatSPN и PIPE2 имеют открытый исходный код (обозначается O), первый по лицензии GPLv2, а второй по лицензии MIT. На использование бинарной версии Oris никаких ограничений нет, т. е. пользователи получают свободный доступ к инструменту (обозначается C), а его Java библиотека Sirio имеет открытый исходный код (обозначается O) по лицензии AGPL.

Заключение. В статье дается обзор инструментальных средств GreatSPN, Oris, PetriNuts, TimeNet и PIPE2, которые предназначены для работы со стохастическими сетевыми моделями и которые многократно применялись для решения различных задач моделирования и анализа сложно организованных систем с вероятностными характеристиками. Для каждого из рассматриваемых инструментов приведены основные его характеристики и возможности. Дано сравнение инструментов по следующим критериям: поддерживаемые классы стохастических моделей, (нестохастические) функции, возможности стохастического анализа, платформы, доступность. Результаты сравнения сведены в таблицу. Статья может быть полезной для исследователей и инженеров при выборе подходящего инструмента для работы со стохастическими моделями.

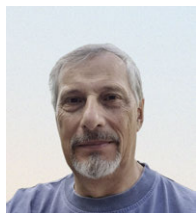
Список литературы

1. Reisig W. Petri Nets: An Introduction. V. 4. Springer, 1985. (EATCS Monographs on Theoretical Computer Science), DOI: 10.1007/978-3-642-69968-9.
2. Boyer M., Roux O. On the Compared Expressiveness of Arc, Place and Transition Time Petri Nets // *Fundamenta Informaticae*. 2008. Jan. V. 88. P. 225–249.
3. Berthomieu B., Diaz M. Modeling and verification of time dependent systems using time Petri nets // *IEEE Transactions on Software Engineering*. 1991. Mar. V. 17, N 3. P. 259–273. DOI: 10.1109/32.75415.
4. Molloy M. Performance Analysis Using Stochastic Petri Nets // *IEEE Trans. Computers*. 1982. V. 31, N 9. P. 913–917. DOI: 10.1109/TC.1982.1676110.
5. Vicario E., Sassoli L., Carnevali L. Using Stochastic State Classes in Quantitative Evaluation of Dense-Time Reactive Systems // *IEEE Trans. Software Eng.* 2009. V. 35, N 5. P. 703–719. DOI: 10.1109/TSE.2009.36.
6. Wang J. Stochastic Timed Petri Nets and Stochastic Petri Nets // *Timed Petri Nets: Theory and Application*. Boston, MA : Springer US, 1998. P. 125–153. DOI: 10.1007/978-1-4615-5537-7_5.
7. Ajmone Marsan M. et al. An introduction to generalized stochastic Petri nets // *Microelectronics Reliability*. 1991. Jan. V. 31, N 4. P. 699–725. DOI: 10.1016/0026-2714(91)90010-5.
8. Ajmone Marsan M., Chiola G. On Petri nets with deterministic and exponentially distributed Bring times // *Advances in Petri Nets 1987*, covers the 7th European Workshop on Applications and Theory of Petri Nets, Oxford, UK, June 1986. V. 266 / ed. by G. Rozenberg. Springer, 1986. P. 132–145. (Lecture Notes in Computer Science), DOI: 10.1007/3-540-18086-9_23.
9. Dugan J. et al. Extended Stochastic Petri Nets: Applications and Analysis // *Performance '84, Proceedings of the Tenth International Symposium on Computer Performance Modelling, Measurement and Evaluation* / ed. by E. Gelenbe. North-Holland, 1984. P. 507–519.
10. Ajmone Marsan M. et al. The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets // *IEEE Trans. Software Eng.* 1989. V. 15, N 7. P. 832–846. DOI: 10.1109/32.29483.
11. German R., Lindemann C. Analysis of stochastic Petri nets by the method of supplementary variables // *Performance Evaluation*. 1994. May. V. 20, N 1–3. P. 317–335. DOI: 10.1016/0166-5316(94)90020-5.
12. Тарасюк И.В. Стохастические сети Петри — формализм для моделирования и анализа производительности вычислительных процессов // *Системная информатика*. Новосибирск, 2004. С. 135–194.
13. German R. Performance analysis of communication systems — modelling with non-Markovian stochastic Petri nets : Modeling with Non-Markovian Stochastic Petri Nets. Wiley, 2000. P. 456.
14. Biagi M. et al. Exploiting Non-deterministic Analysis in the Integration of Transient Solution Techniques for Markov Regenerative Processes // *Quantitative Evaluation of Systems*. Springer International Publishing, 2017. P. 20–35. DOI: 10.1007/978-3-319-66335-7_2.
15. Martina S. et al. Performance Evaluation of Fischer's Protocol through SteadyState Analysis of Markov Regenerative Processes // *IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 09/2016. P. 355–360. DOI: 10.1109/MASCOTS.2016.72.
16. Horvath A. et al. Transient analysis of non-Markovian models using stochastic state classes // *Performance Evaluation*. 2012. V. 69, N 7/8. P. 315–335. DOI: 10.1016/j.peva.2011.11.002.
17. Amparore E. Stochastic Modelling and Evaluation Using GreatSPN // *ACM- SIGMETRICS Performance Evaluation Review*. New York, NY, USA, 2022. June. V. 49, N 4. P. 87–91. DOI: 10.1145/3543146.3543165.

18. Amparore E. et al. Years of GreatSPN // Principles of Performance and Reliability Modeling and Evaluation: Essays in Honor of Kishor Trivedi on his 70th Birthday / ed. by L. Fiondella, A. Puliafito. Cham : Springer International Publishing, 2016. P. 227–254. DOI: 10.1007/978-3-319-30599-8-9.
19. K. J., Kristensen L. Coloured Petri Nets. Springer Berlin Heidelberg, 2009. DOI: 10.1007/b95112.
20. ISO/IEC. Software and Systems Engineering - High-level Petri Nets, Part 2: Transfer Format, International Standard ISO/IEC 15909, February 2011.
21. Kindler E. The Petri Net Markup Language and ISO/IEC 15909-2: Concepts, Status, and Future Directions // Tagungsband Entwurf komplexer Automatisierungssysteme EKA. 2006. P. 35–55.
22. Clarke E., Emerson E. Design and synthesis of synchronization skeletons using branching time temporal logic // Logics of Programs / ed. by D. Kozen. Springer Berlin Heidelberg, 1982. P. 52–71.
23. Deharbe D. A Tutorial Introduction to Symbolic Model Checking // Logic for Concurrency and Synchronisation / ed. by R. de Queiroz. Dordrecht : Springer Netherlands, 2003. P. 215–237. DOI: 10.1007/0-306-48088-3_5.
24. Beccuti M., Franceschinis G., Haddad S. Markov Decision Petri Net and Markov Decision Well-Formed Net Formalisms // Petri Nets and Other Models of Concurrency — ICATPN 2007 / ed. by J. Kleijn, A. Yakovlev. Springer Berlin Heidelberg, 2007. P. 43–62. DOI: 10.1007/978-3-540-73094-1_6.
25. Emerson E., Sistla A. Symmetry and model checking // Formal Methods in System Design. 1996. V. 9, N 1. P. 105–131. DOI: 10.1007/BF00625970.
26. Babar J. et al. GrcatSPN Enhanced with Decision Diagram Data Structures // Applications and Theory of Petri Nets / ed. by J. Lilius, W. Penczek. Springer Berlin Heidelberg, 2010. P. 308–317.
27. Chaki S., Gurfinkel A. BDD-Based Symbolic Model Checking // Handbook of Model Checking / ed. by E. Clarke et al. Springer, 2018. P. 219–245. DOI: 10.1007/978-3-319-10575-8_8.
28. R. R., S. B., Zimmermann A. An Evaluation Framework for Comparative Analysis of Generalized Stochastic Petri Net Simulation Techniques // IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2020. V. 50. P. 2834–2844. DOI: 10.1109/TSMC.2018.2837643.
29. Pernice S. et al. Multiple Sclerosis Disease: A Computational Approach for Investigating Its Drug Interactions // Computational Intelligence Methods for Bioinformatics and Biostatistics / ed. by P. Cazzaniga et al. Cham : Springer International Publishing, 2020. P. 299–308. DOI: 10.1007/978-3-030-63061-4_26.
30. Amparore E., Donatelli S., Landini E. Modelling and Evaluation of a Control Room Application // Application and Theory of Petri Nets and Concurrency / ed. by W. van der Aalst, E. Best. Cham: Springer International Publishing, 2017. P. 243–263.
31. Richard L. Performance Results for the CSMA/CD Protocol Using GrcatSPN // Journal of Systems and Software. 1997. V. 37, N 1. P. 75–90. DOI: 10.1016/S0164-1212(96)00041-6.
32. Amparore E., Donatelli S. GreatTeach: A Tool for Teaching (Stochastic) Petri Nets // Application and Theory of Petri Nets and Concurrency. Springer International Publishing, 2018. P. 416–425. DOI: 10.1007/978-3-319-91268-4_24.
33. Castagno P. et al. A computational framework for modeling and studying pertussis epidemiology and vaccination // BMC Bioinformatics. 2020. V. 21, N 8. P. 344. DOI: 10.1186/S12859-020-03648-6.
34. The GrcatSPN Framework. [El. Res.]: <https://github.com/greatspn/SOURCES>. Accessed: 2024-03-15.
35. Paolieri M. et al. The ORIS Tool: Quantitative Evaluation of Non-Markovian Systems // IEEE Trans. Software Eng. 2021. V. 47, N 6. P. 1211–1225. DOI: 10.1109/TSE.2019.2917202.
36. Carnevali L., Paolieri M., Vicario E. The ORIS tool: app, library, and toolkit for quantitative evaluation of non-Markovian systems // ACM SIGMETRICS Performance Evaluation Review. 2022. V. 49, N 4. P. 81–86. DOI: 10.1145/3543146.3543164.

37. Stewart W. Introduction to the Numerical Solution of Markov Chains. Princeton University Press, 1995. DOI: 10.1515/9780691223384.
38. Carnevali L. et al. Non-Markovian Performability Evaluation of ERTMS/ETCS Level 3 // Computer Performance Engineering - 12th European Workshop, EPEW 2015. V. 9272 / ed. by M. Beltran, W. Knottenbelt, J. Bradley. Cham : Springer, 2015. P. 47–62. (Lecture Notes in Computer Science), DOI: 10.1007/978-3-319-23267-6_4.
39. Biagi M. et al. Model-Based Quantitative Evaluation of Repair Procedures in Gas Distribution Networks // ACM Trans. Cyber Phys. Syst. 2019. V. 3, N 2. 19:1–19:26. DOI: 10.1145/3284037.
40. Carnevali L., Tarani F., Vicario F. Performability Evaluation of Water Distribution Systems During Maintenance Procedures // IEEE Trans. Syst. Man Cybern. Syst. 2020. V. 50, N 5. P. 1704–1720. DOI: 10.1109/TSMC.2017.2783188.
41. Carnevali L. et al. Using the ORIS Tool and the SIRIO Library for Model- Driven Engineering of Quantitative Analytics // Computer Performance Engineering / ed. by K. Gilly, N. Thomas. Cham: Springer International Publishing, 2023. P. 200–215. DOI: 10.1007/978-3-031-25049-1_13.
42. ORIS Tool. [El. Res.]: <http://www.oris-tool.org>. Accessed: 2024-03-15.
43. ORIS Tool: The Sirio Library. [El. Res.]: <https://github.com/oris-tool/sirio>. Accessed: 2024-03-15.
44. Heiner M. et al. Snoopy — A Unifying Petri Net Tool // Application and Theory of Petri Nets. PETRI NETS 2012. V. 7347 / ed. by S. Haddad, L. Pomello. Springer Berlin Heidelberg, 2012. P. 398–407. (Lecture Notes in Computer Science), DOI: 10.1007/978-3-642-31131-4_22.
45. David R., Alla H. Discrete, Continuous, and Hybrid Petri Nets. Springer Berlin Heidelberg, 2010. P. 550. DOI: 10.1007/978-3-642-10669-9.
46. Liu F., Heiner M., Gilbert D. Fuzzy Petri nets for modelling of uncertain biological systems // Briefings in Bioinformatics. 2018. Dec. V. 21, N 1. P. 198–210. DOI: 10.1093/bib/bbyl18.
47. Fujita M., McGeer P., Yang J. Multi-Terminal Binary Decision Diagrams: An Efficient DataStructure for Matrix Representation // Form. Methods Syst. Des. USA, 1997. Apr. V. 10, N 2/3. P. 149–169. DOI: 10.1023/A: 1008647823331.
48. Hucka M. et al. Systems Biology Markup Language (SBML) Level 2 Version 5: Structures and Facilities for Model Definitions // Journal of Integrative Bioinformatics. 2015. V. 12. N 2. P. 731–901. DOI: 10.2390/biecoll-jib-2015-271.
49. Heiner M., Schwarick M., Wegener J.-T. Charlie — An Extensible Petri Net Analysis Tool // Application and Theory of Petri Nets and Concurrency / ed. by R. Devillers, A. Valmari. Cham : Springer International Publishing, 2015. P. 200–211. DOI: 10.1007/978-3-319-19488-2_10.
50. Heiner M., Rohr C., Schwarick M. MARGIE — Model Checking and Reachability Analysis Done Efficiently // Application and Theory of Petri Nets and Concurrency / ed. by J.-M. Colom, J. Desel. Springer Berlin Heidelberg, 2013. P. 389–399. DOI: 10.1007/978-3-642-38697-8_21.
51. Baier C. et al. Model Checking Continuous-Time Markov Chains by Transient Analysis // Computer Aided Verification / ed. by E. Emerson, A. Sistla. Springer Berlin Heidelberg, 2000. P. 358–372.
52. Donaldson R., Gilbert D. A Model Checking Approach to the Parameter Estimation of Biochemical Pathways // Computational Methods in Systems Biology / ed. by M. Heiner, A. M. Uhrmacher. Springer Berlin Heidelberg, 2008. P. 269–287.
53. Chodak J., Heiner M. Spike — Reproducible Simulation Experiments with Configuration File Branching // Computational Methods in Systems Biology. Springer International Publishing, 2019. P. 315–321. DOI: 10.1007/978-3-030-31304-3_19.
54. Gilbert D., Donaldson R. A Monte Carlo model checker for probabilistic LTL with numerical constraints : tech. reP. / Bioinformatics Research Centre, University of Glasgow. 01/2008.

55. Gilbert D. et al. Spatial quorum sensing modelling using coloured hybrid Petri nets and simulative model checking // BMC Bioinformatics. 2019. V. 20, supplement 4. DOI: 10.1186/sl2859-019-2690-z.
56. Hecrajy M. et al. Snoopy's hybrid simulator: a tool to construct and simulate hybrid biological models // BMC Systems Biology. 2017. July. V. 11, N 1. DOI: 10.1186/S12918-017-0449-6.
57. Zimmermann A. Modelling and Performance Evaluation with TimeNET 4.4 // Quantitative Evaluation of Systems - 14th International Conference, QEST 2017. V. 10503 / ed. by N. Bertrand, L. Bortolussi. Springer, 2017. P. 300–303. (Lecture Notes in Computer Science), DOI: 10.1007/978-3-319-66335-7_19.
58. Selic B. Modeling And Analysis Of Realtime And Embedded Systems With Uml And Marte Developing Cyberphysical Systems. Elsevier Science & Technology, 2014. DOI: 10.1016/C2012-0-13536-5.
59. Zimmermann A. et al. Analysis of Safety-Critical Cloud Architectures with MultiTrajectory Simulation // Annual Reliability and Maintainability Symposium (RAMS). 01/2022. P. 1–7. DOI: 10.1109/RAMS51457.2022.9893923.
60. Fedorova A., Beliautsov V., Zimmermann A. Colored Petri Net Modelling and Evaluation of Drone Inspection Methods for Distribution Networks // Sensors. 2022. V. 22, N 9. DOI: 10.3390/s22093418.
61. Dingle N., Knottenbelt W., Suto T. PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets // SIGMETRICS Performance Evaluation Review ACM. New York, NY, USA, 2009. Mar. V. 36, N 4. P. 34–39. DOI: 10.1145/1530873.1530881.
62. Platform Independent Petri Net Editor v4. [El. Res.]: <https://sourceforge.net/projects/pipe2>. Accessed: 2024-03-15.
63. PIPE 5. [El. Res.]: <https://github.com/sarahtattersall/PIPE>. Accessed: 2024-03-15.



Быстров Александр Васильевич — старший научный сотрудник Института систем информатики им. А.П. Ершова СО РАН, доцент кафедры «Системы информати-

ки» Новосибирского государственного университета. E-mail: avb@iis.nsk.su. Кандидат физ.-мат. наук с 2008 г. Автор и соавтор более 40 научных работ. Основные научные интересы — спецификация и верификация параллельных систем и систем реального времени и соответствующие программные средства и языки программирования.

Bystrov Alexandr Vasilievich graduated Novosibirsk State University, Ph.D. (Candidate of science) degree since 2008, senior researcher at the A.P. Ershov Institute of Informatics Systems SB RAS and associate professor at the Novosibirsk State University. Author of more than 40 scientific papers. Research interests: specification and verification of concurrent/real-time systems, related software tools and languages.



Вирбицкайте Ирина Бонавентуровна — главный научный сотрудник, зав. лабораторией «Теории параллельных процессов» Института систем информатики им. А.П. Ершова

СО РАН, профессор Новосибирского государственного университета. E-mail: virb@iis.nsk.su. Ученые степени кандидата (1990) и доктора (2002) физико-математических наук, ученые звания старшего научного сотрудника (1995), профессора (2008) по специальности «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей». Автор и соавтор более 150 научных трудов. Основные направления научной деятельности — теория параллельных систем и процессов, включая формальные модели параллелизма, спецификацию и верификацию параллельных систем и систем реального времени, автоматическое конструирование параллельных программ.

Virbitskaite Irina Bonaventurovna is a chief researcher, the head of the laboratory

“Theory of Concurrent Processes” at the A.P. Ershov Institute of Informatics Systems SB RAS and professor at Novosibirsk State University. Academic degrees of Candidate (1990) and Doctor (2002) of Physical and Mathematical Sciences, academic titles of senior researcher (1995), professor (2008), in the specialty “Mathematical and software support of computers, complexes and computer networks”. Author and co-author of more than 150 scientific papers. Research interests: theory of concurrent systems and processes, including formal models for concurrency, specification and verification of concurrent and real-time systems, automatic design of parallel programs.



Елена Сергеевна Ошевская — младший научный сотрудник Института систем информатики им. А.П. Ершова (Новосибирск). E-mail: eso@iis.nsk.su. В 2005 г. закончила Механико-математический

факультет Новосибирского государственного университета по направлению «Математика». В 2013 г. получила ученую степень кандидата физико-математических наук, защитив диссертацию на тему «Категорные методы исследования полукубических множеств и пространств как моделей параллельных процессов». Область научных интересов: спецификация и верификация параллельных систем и систем реального времени.

Elena Sergeevna Oshevskaya is a junior researcher at A.P. Ershov Institute of Informatics Systems (Novosibirsk city). Graduated from the Faculty of Mechanics and Mathematics of Novosibirsk State University in the field of Mathematics (2005), academic degree of Candidate of Physical and Mathematical Sciences, having defended her dissertation on the topic “Categorical methods in studying semicubic sets and spaces as models of concurrent processes”. Research interests: specification and verification of concurrent and real-time systems.

Дата поступления — 19.03.2024