System Modelling, Validation and Testing using Kernel P Systems

Marian Gheorghe[1], Florentin Ipate[2]

[1] School of Electrical Engineering and Computer Science,
University of Bradford, West Yorkshire, Bradford BD7 1DP, UK
`m.gheorghe@bradford.ac.uk`
[2] Department of Computer Science, Faculty of Mathematics and Computer Science,
University of Bucharest, Str. Academiei nr. 14, 010014, Bucharest, Romania
`florentin.ipate@ifsoft.ro`

Abstract. Nature inspired computational approaches have been on the focus of researchers for several decades. Membrane computing [14] is one of these paradigms that has recently been through significant developments leading to a broad spectrum of outcomes. The main computational models are called membrane systems or *P systems* and are inspired by the functioning and structure of the living cells as well as of more complex higher-oder biological entities, such as tissues and organs.

In recent years, various types or classes of P systems have been introduced and and studied, and in certain cases applied to different problems. While these variants provide more flexibility in modelling, this has inevitably resulted in a large pool of P system variants, which do not have a coherent integrating view and might be a drawback in analysing these models with some standard formal verification methods and tools.

Kernel P (kP) systems have been introduced to unify many variants of P system models, and combine a blend of various P system features and concepts, including (i) complex guards attached to rules, (ii) flexible ways to specify the system structure and dynamically change it and (iii) various execution strategies for rules and compartments.

The usability and efficiency of kP systems have been illustrated by a number of representative case studies, ranging from systems and synthetic biology, e.g. quorum sensing [12], genetic Boolean gates [15] and synthetic pulse generators [1], to some classical computational problems, e.g. sorting [6], broadcasting [9] and subset sum [5].

Kernel P system models are supported by an integrated software suite, kPWorkbench, which employs a set of simulation and formal verification tools and methods that permit simulating and verifying them. The models are expressed in a specification language, called *kP-Lingua*. The verification component of kPWorkbench [5] checks the correctness of kP system models by exhaustively analysing all possible behaviours. In order to facilitate the specification of system requirements, kPWorkbench features a property language, called *kP-Queries*, which comprises a list of property patterns written as natural language statements. The properties expressed in *kP-Queries* are verified using the Spin [11] and NuSMV [3] model checkers after being translated into corresponding *Linear Temporal Logic (LTL)* and *Computation Tree Logic*

(CTL) syntax. The simulation component features a native simulator [2, 13], which allows the users to simulate kP system models efficiently. In addition, kPWorkbench integrates the Flame simulator [4, 15], a general purpose large scale agent based simulation environment, based on a method that allows users to express kP systems as a set of communicating X-machines [10].

Significant progress has also been made in the area of testing applications modelled by kP systems. When testing a kP system model, an automata model needs to be constructed first, based on the computation tree of the kP system. As, in general, the computation tree may be infinite and cannot be modelled by a finite automaton, an approximation of the tree is used. This approximation is obtained by limiting the length of any computation to an upper bound k and considering only computations up to k transitions in length. This approximation is then used to construct a deterministic finite cover automaton (DFCA) of the model [6–8].

This paper reviews the main achievements in the area of kP systems modelling, verification and testing.

References

1. Bakir, M.E., Ipate, F., Konur, S., Mierla, L., Niculescu, I.: Extended simulation and verification platform for kernel P systems. In: Gheorghe, M., Rozenberg, G., Salomaa, A., Sosík, P., Zandron, C. (eds.) Membrane Computing - 15th International Conference, CMC 2014, Prague, Czech Republic, August 20-22, 2014, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8961, pp. 158–178. Springer (2014), `http://dx.doi.org/10.1007/978-3-319-14370-5_10`
2. Bakir, M.E., Konur, S., Gheorghe, M., Niculescu, I., Ipate, F.: High performance simulations of kernel P systems. In: 2014 IEEE International Conference on High Performance Computing and Communications, 6th IEEE International Symposium on Cyberspace Safety and Security, 11th IEEE International Conference on Embedded Software and Systems, HPCC/CSS/ICESS 2014, Paris, France, August 20-22, 2014. pp. 409–412 (2014), `http://dx.doi.org/10.1109/HPCC.2014.69`
3. Cimatti, A., Clarke, E.M., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A.: NuSMV 2: An opensource tool for symbolic model checking. In: Brinksma, E., Larsen, K.G. (eds.) Computer Aided Verification, 14th International Conference, CAV 2002, Copenhagen, Denmark, July 27-31, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2404, pp. 359–364. Springer (2002), `http://dx.doi.org/10.1007/3-540-45657-0_29`
4. Coakley, S., Gheorghe, M., Holcombe, M., Chin, S., Worth, D., Greenough, C.: Exploitation of high performance computing in the FLAME agent-based simulation framework. In: Min, G., Hu, J., Liu, L.C., Yang, L.T., Seelam, S., Lefèvre, L. (eds.) 14th IEEE International Conference on High Performance Computing and Communication & 9th IEEE International Conference on Embedded Software and Systems, HPCC-ICESS 2012, Liverpool, United Kingdom, June 25-27, 2012. pp. 538–545. IEEE Computer Society (2012), `http://dx.doi.org/10.1109/HPCC.2012.79`
5. Dragomir, C., Ipate, F., Konur, S., Lefticaru, R., Mierla, L.: Model checking kernel P systems. In: Alhazov, A., Cojocaru, S., Gheorghe, M., Rogozhin, Y., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing - 14th International Conference, CMC 2013, Chişinău, Republic of Moldova, August 20-23, 2013, Revised Se-

lected Papers. Lecture Notes in Computer Science, vol. 8340, pp. 151–172. Springer (2013), `http://dx.doi.org/10.1007/978-3-642-54239-8_12`

6. Gheorghe, M., Ceterchi, R., Ipate, F., Konur, S.: Kernel P systems modelling, testing and verification - sorting case study. In: Leporati, A., Rozenberg, G., Salomaa, A., Zandron, C. (eds.) Membrane Computing - 17th International Conference, CMC 2016, Milan, Italy, July 25-29, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10105, pp. 233–250. Springer (2016), `http://dx.doi.org/10.1007/978-3-319-54072-6_15`
7. Gheorghe, M., Ceterchi, R., Ipate, F., Konur, S., Lefticaru, R.: Kernel P systems: from modelling to verification and testing. Theoretical Computer Science (accepted for publication), `http://hdl.handle.net/10454/11720`
8. Gheorghe, M., Ipate, F.: On testing P systems. In: Corne, D.W., Frisco, P., Paun, G., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing - 9th International Workshop, WMC 2008, Edinburgh, UK, July 28-31, 2008, Revised Selected and Invited Papers. Lecture Notes in Computer Science, vol. 5391, pp. 204–216. Springer (2008), `http://dx.doi.org/10.1007/978-3-540-95885-7_15`
9. Gheorghe, M., Konur, S., Ipate, F., Mierla, L., Bakir, M.E., Stannett, M.: An integrated model checking toolset for kernel P systems. In: Rozenberg, G., Salomaa, A., Sempere, J.M., Zandron, C. (eds.) Membrane Computing - 16th International Conference, CMC 2015, Valencia, Spain, August 17-21, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9504, pp. 153–170. Springer (2015), `http://dx.doi.org/10.1007/978-3-319-28475-0_11`
10. Holcombe, M.: X-machines as a basis for dynamic system specification. Software Engineering Journal 3(2), 69–76 (1988), `http://dx.doi.org/10.1049/sej.1988.0009`
11. Holzmann, G.J.: The model checker SPIN. IEEE Transactions on Software Engineering 23(5), 275–295 (1997)
12. Konur, S., Gheorghe, M., Dragomir, C., Mierla, L., Ipate, F., Krasnogor, N.: Qualitative and quantitative analysis of systems and synthetic biology constructs using P systems. ACS Synthetic Biology 4(1), 83–92 (2015), `http://dx.doi.org/10.1021/sb500134w`, pMID: 25090609
13. Konur, S., Kiran, M., Gheorghe, M., Burkitt, M., Ipate, F.: Agent-based high-performance simulation of biological systems on the GPU. In: 17th IEEE International Conference on High Performance Computing and Communications, HPCC 2015, 7th IEEE International Symposium on Cyberspace Safety and Security, CSS 2015, and 12th IEEE International Conference on Embedded Software and Systems, ICESS 2015, New York, NY, USA, August 24-26, 2015. pp. 84–89. IEEE (2015), `http://dx.doi.org/10.1109/HPCC-CSS-ICESS.2015.253`
14. Păun, G.: Computing with membranes. Journal of Computer and System Sciences 61(1), 108–143 (2000), `http://dx.doi.org/10.1006/jcss.1999.1693`
15. Sanassy, D., Fellermann, H., Krasnogor, N., Konur, S., Mierla, L., Gheorghe, M., Ladroue, C., Kalvala, S.: Modelling and stochastic simulation of synthetic biological boolean gates. In: 2014 IEEE International Conference on High Performance Computing and Communications, 6th IEEE International Symposium on Cyberspace Safety and Security, 11th IEEE International Conference on Embedded Software and Systems, HPCC/CSS/ICESS 2014, Paris, France, August 20-22, 2014. pp. 404–408 (2014), `http://dx.doi.org/10.1109/HPCC.2014.68`