Reactics: Model Checker for Distributed Reaction Systems

Marcin Piątkowski
Nicolaus Copernicus University, Toruń, Poland
marcin.piatkowski@mat.umk.pl

Artur Męski
ICS, Polish Academy of Sciences, Warsaw, Poland
meski@ipipan.waw.pl

Łukasz Mikulski
Nicolaus Copernicus University, Toruń, Poland
lukasz.mikulski@mat.umk.pl

Wojciech Penczek
ICS, Polish Academy of Sciences, Warsaw, Poland
penczek@ipipan.waw.pl

ABSTRACT

Reaction systems provide a qualitative formalism for modeling cellular behavior through facilitation and inhibition. We present an extension of the tool Reactics supporting distributed reaction systems and enabling the model checking of temporal-epistemic properties using the rsctlk logic and a BDD-based verification engine.

KEYWORDS

Reaction Systems, Model Checking, Temporal-Epistemic Logics

ACM Reference Format:
Marcin Piątkowski, Artur Męski, Łukasz Mikulski, and Wojciech Penczek. 2026. Reactics: Model Checker for Distributed Reaction Systems. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/QOBE7447

1 INTRODUCTION

Natural computing investigates computational models inspired by natural processes and information processing in nature. A key area within this field, bio-molecular computation, explores biological processes as computational phenomena, focusing on interactions between biochemical reactions governed by facilitation and inhibition [34]. Reaction systems [9, 14, 15, 25–29, 35] provide a qualitative formalism in this domain [13]. This paper presents a tool for verifying distributed (multi-agent) reaction systems [36].

Challenges. Model checking for distributed reaction systems with temporal-epistemic properties requires managing a complex space of interactions and knowledge states [20, 21, 30], and was proven PSPACE-complete [5, 18, 40]. supporting model checking for rsctlk [36, 37] (a logic that combines temporal and epistemic operators) implemented using BDDs [1]. To the best of our knowledge, this is the only model checker supporting such verification.

2 RELATED WORK

In recent years, several tools have been developed to simulate reaction systems, including HERESY [39], cl-rs [17], WebRSim [19], and brsim [6]. More recent tools have adopted reaction-system–based analytical frameworks for applications in systems biology [7, 8, 10], with special attention devoted to a simulator BioResolve [11]. Among general-purpose model-checking tools, the most prominent ones supporting multi-agent systems with epistemic modalities MCMAS [33], Verics [23], MCK [3], Uppaal [32], PRISM [31], STV [24], and VITAMIN [16]. However, none of these tools supports semantics that closely match reaction systems. Therefore, their use for verifying reaction systems suffers from an exponential growth in complexity, which is demonstrated for MCMAS in [36].

3 APPLICATION DOMAIN

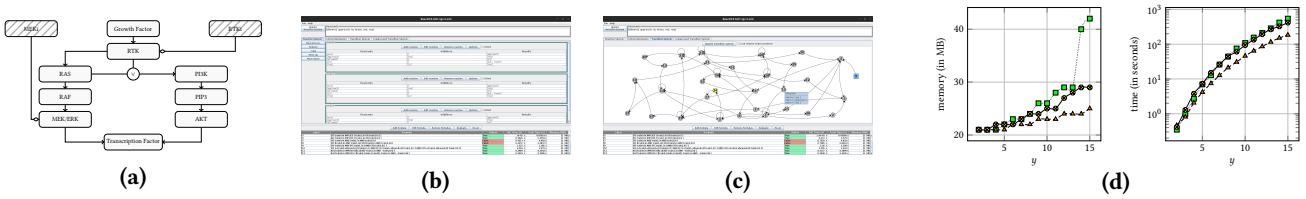
Distributed reaction systems provide a flexible framework for modeling complex biological processes involving multiple interacting agents with individual dynamics [13, 36]. Typical examples include tumor immune system interactions, bacterial competition in microbiomes, and immune system coordination among different cell types [12]. A key advantage of the proposed approach is its support for modular modeling. Individual components can be developed independently and later integrated into a unified system, with their interactions mediated by the distributed environment. This environment can be used to control agent activity across different stages of the modeled process. Our tool enables integrated inspection of system evolution and agents' knowledge states. We demonstrate the tool in action on a benchmark of the signaling transduction pathways in breast cancer motivated by [43].

4 FORMAL BACKGROUND AND LOGIC

Distributed reaction systems (DRS) are a structure consisting of multiple agents, each having their own reactions, and operating over a shared set of entities [36]. A reaction consists of three finite sets of entities: *reactants*, *inhibitors*, and *products*. It is enabled when all reactants are present and all inhibitors are absent; when enabled, the reaction produces all entities in its product set [13]. The global states of a system (tuples of agents' local states) are defined by finite sets of entities and evolve through predefined reactions [38]. An evolution (an *interactive process*) proceeds stepwise, where at each step, all active agents apply their local reactions in parallel and their results are combined to produce the global system state. Inactive agents retain their state until they become active again. We use a *context-automaton* that produces a context sequence, selecting the active agents and optionally providing them with additional entities. A model M for DRS is a product of the global state space and a context automaton. The paths correspond to interactive processes.

The *computation tree logic of knowledge for reaction systems* [36] (rsctlk) is an extension of rsctl [38] with epistemic operators. Its formulas use the propositional and temporal operators of rsctl,

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/QOBE7447



(a) (b) (c) (d)

Figure 1: The model of signal transduction (a) and screenshots with corresponding DRS (b) and resulting transition system (c). The graphs of the operation time and memory used versus time for x equal to 2 (□), 3 (△) and 4 (⊗) (d).

such as next (X), globally (G), and until (U). Each operator is preceded by the existential or universal path quantifier ($\mathsf{E}_{\mathfrak{sc}}$, $\mathsf{A}_{\mathfrak{sc}}$) that can be restricted by a state constraint ($\mathfrak{sc}$). The state constraint limits the admissible contexts (intuitively, actions) over which path quantification ranges. Instead of standard propositional variables, the logic uses atoms of the form $i.ent$ describing the presence of entity ent in the local state of individual agent i. The logic also incorporates epistemic operators for expressing knowledge, and it is interpreted in models M for distributed reaction systems. The formula $\mathrm{K}_i\phi$ states that agent i knows ϕ. Formally, $M, s \models \mathrm{K}_i\phi$ if $M, s' \models \phi$ for all global states s' in M sharing the same i-local state of s. For example, $M, s \models \mathsf{A}_{sc}\mathrm{G}(\mathrm{K}_0\phi)$ if for all states at all paths in M starting at s and constrained by sc we have that agent 0 knows that ϕ holds.

5 DESIGN AND IMPLEMENTATION

ReactICS is a comprehensive open-source toolkit for modelling and verifying reaction systems, implemented in C++. The original algorithms for rsCTL were introduced in [38]. The core functionality supporting rsCTLK was implemented within a doctoral dissertation [4] and has not been published elsewhere. ReactICS comprises two main modules: a BDD-based module for compact representation and efficient manipulation of the system state space, and an SMT-based module that encodes verification problems as satisfiability modulo theories. In addition, it supports symbolic model checking for rsCTLK, allowing the formal verification of systems where both temporal and epistemic aspects are crucial. Recently, ReactICS has been extended with a Java-based graphical user interface (GUI) that uses the JUNG 2.0 library [22] for graph visualisation. The system editor module enables users to define and modify the structure of a distributed reaction system by specifying its processes (or agents) and their associated reactions. The context automaton editor module provides graphical support for editing the context automaton, including states, transitions, guards, and agent-specific context. The state space visualisation module allows users to inspect the system behaviour as a transition system, either in its full form or in a compressed representation that abstracts from context automaton states. The verification module enables direct verification of system properties expressed in rsCTLK via the graphical interface.

6 EXAMPLE SCENARIO

As an example application, we consider a model of simplified intracellular signal transduction in breast cancer from [36]. The signalling process is represented using distributed reaction systems, which capture the discrete and asynchronous dynamics of interacting signalling pathways. The model describes a signalling network composed of growth factors, receptor tyrosine kinases, signalling proteins, enzymes, and transcription factors, as shown in Figure 1(a). Each component can be either present (activated) or absent (deactivated). Drug inhibition is explicitly modelled by introducing inhibitors.

The system evolution is formalised as an interactive process in which agents apply their local reactions in parallel at each step, and the global state is obtained by combining their local results under the influence of a shared context. This example demonstrates how logical regulatory rules of a signalling network [2] can be systematically translated into distributed reaction systems, and how activation sequences of agents lead to system evolutions reported in [43], from which the original motivation was derived.

7 EXPERIMENTAL EVALUATIONS

For the experimental evaluation, we used a benchmark based on the example from Section 6, maintaining the same pathway structure with a common beginning and end. We considered the scaling parameter x between 2 and 4 for the number of modules, and y between 2 and 15 for the process length. We evaluated the formulas $\phi_y = \mathsf{A}_{sc}\mathrm{G}(\mathrm{K}_0(\bigvee_{i=1}^{y} i.\mathrm{TF}) \vee \mathrm{K}_0(\bigwedge_{i=1}^{y} \neg i.\mathrm{TF}))$, where $sc = \bigwedge_{i=1}^{y} i.GF$, expressing that the treatment agent 0 knows whether the proposed therapy is effective. Here, i.GF indicates that the growth factor is overactive for agent i, while i.TF indicates that agent i has produced the entity TF, implying the therapy was not successful. The experiments were conducted on a machine equipped with an Intel® Xeon® Platinum 8260 processor and 1 TB of RAM, running Debian Linux. The resulting time and memory consumption for model checking ϕ_y are shown in Fig. 1 (d).

8 CONCLUSIONS

Unlike existing approaches focused on simulation, ReactICS enables formal verification of both system dynamics and agents' knowledge using the rsCTLK logic. The tool is complemented by an intuitive graphical user interface that supports modular modelling, inspection of state spaces, and verification of properties without requiring in-depth expertise in formal methods. A biologically motivated case study demonstrates that ReactICS faithfully captures asynchronous signalling behaviour while allowing exhaustive analysis of all possible executions.

The complete system and example input files are available at [41]; a short video presentation can be found at [42].

REFERENCES

[1] Akers. 1978. Binary decision diagrams. *IEEE Transactions on computers* 100, 6 (1978), 509–516.

[2] Reka Albert and Juilee Thakar. 2014. Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 6, 5 (2014), 353–369.

[3] Kaya Alpturer, Gerald Huang, and Ron van der Meyden. 2025. Model Checking and Synthesis for Optimal Use of Knowledge in Consensus Protocols. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '25)*. 131–140.

[4] Męski Artur. 2019. *Model Checking for Reaction and Multi-Agent Systems.* Ph.D. Dissertation. Institute of Computer Science, Polish Academy of Sciences.

[5] Sepinoud Azimi, Cristian Gratie, Sergiu Ivanov, Luca Manzoni, Ion Petre, and Antonio E Porreca. 2016. Complexity of model checking for reaction systems. *Theoretical Computer Science* 623 (2016), 103–113.

[6] Sepinoud Azimi, Cristian Gratie, Sergiu Ivanov, and Ion Petre. 2015. Dependency graphs and mass conservation in reaction systems. *Theoretical Computer Science* 598 (2015), 23–39.

[7] Demis Ballis, Linda Brodo, and Moreno Falaschi. 2024. Modeling and analyzing reaction systems in Maude. *Electronics* 13, 6 (2024), 1139.

[8] Demis Ballis, Linda Brodo, Moreno Falaschi, and Carlos Olarte. 2024. Process calculi and rewriting techniques for analyzing reaction systems. In *International Conference on Computational Methods in Systems Biology*. Springer, 1–18.

[9] Robert Brijder, Andrzej Ehrenfeucht, Michael Main, and Grzegorz Rozenberg. 2011. A tour of reaction systems. *International Journal of Foundations of Computer Science* 22, 07 (2011), 1499–1517.

[10] Linda Brodo, Roberto Bruni, and Moreno Falaschi. 2021. A logical and graphical framework for reaction systems. *Theoretical Computer Science* 875 (2021), 1–27.

[11] Linda Brodo, Roberto Bruni, Moreno Falaschi, and Ion Petre. 2025. Simulation and Analysis of Distributed Reaction Systems. *IEEE Access* (2025).

[12] Luca Corolli, Carlo Maj, Fabrizio Marini, Daniela Besozzi, and Giancarlo Mauri. 2012. An excursion in reaction systems: From computer science to biology. *Theoretical computer science* 454 (2012), 95–108.

[13] Andrzej Ehrenfeucht, Jetty Kleijn, Maciej Koutny, and Grzegorz Rozenberg. 2012. Reaction Systems: A Natural Computing Approach to the Functioning of Living Cells. *A Computable Universe, Understanding and Exploring Nature as Computation (H. Zenil, ed.)* (2012), 189–208.

[14] Andrzej Ehrenfeucht and Grzegorz Rozenberg. 2007. Events and modules in reaction systems. *Theoretical Computer Science* 376, 1-2 (2007), 3–16.

[15] Andrzej Ehrenfeucht and Grzegorz Rozenberg. 2007. Reaction systems. *Fundamenta informaticae* 75, 1-4 (2007), 263–280.

[16] Aangelo Ferrando and Vadim Malvone. 2025. VITAMIN: A Compositional Framework for Model Checking of Multi-Agent Systems.. In *17th International Conference on Agents and Artificial Intelligence (ICAART 2025)*. SciTePress, 648–655.

[17] Claudio Ferretti, Alberto Leporati, Luca Manzoni, and Antonio E Porreca. 2020. The many roads to the simulation of reaction systems. *Fundamenta Informaticae* 171, 1-4 (2020), 175–188.

[18] Enrico Formenti, Luca Manzoni, and Antonio E Porreca. 2015. On the complexity of occurrence and convergence problems in reaction systems. *Natural Computing* 14 (2015), 185–191.

[19] Sergiu Ivanov, Vladimir Rogojin, Sepinoud Azimi, and Ion Petre. 2018. Webrsim: A web-based reaction systems simulator. *Enjoying Natural Computing: Essays Dedicated to Mario de Jesús Pérez-Jiménez on the Occasion of His 70th Birthday* (2018), 170–181.

[20] Wojciech Jamroga, Michał Knapik, Damian Kurpiewski, and Łukasz Mikulski. 2019. Approximate verification of strategic abilities under imperfect information. *Artificial Intelligence* 277 (2019), 103172.

[21] Wojciech Jamroga, Wojciech Penczek, Teofil Sidoruk, Piotr Dembiński, and Antoni Mazurkiewicz. 2020. Towards partial order reductions for strategic ability. *Journal of Artificial Intelligence Research* 68 (2020), 817–850.

[22] JUNG 2.0 2016. https://jung.sourceforge.net/.

[23] Magdalena Kacprzak, Wojciech Nabiałek, Artur Niewiadomski, Wojciech Penczek, Agata Półrola, Maciej Szreter, Bożena Woźna, and Andrzej Zbrzezny. 2008. Verics 2007-a model checker for knowledge and real-time. *Fundamenta Informaticae* 85, 1-4 (2008), 313–328.

[24] Mateusz Kamiński, Damian Kurpiewski, and Wojciech Jamroga. 2024. STV+KH: Towards Practical Verification of Strategic Ability for Knowledge and Information Flow. In *23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024)*. 2812–2814.

[25] Mariusz Kaniecki and Łukasz Mikulski. 2024. On categorical approach to reaction systems: M. Kaniecki, Ł. Mikulski. *Natural Computing* 23, 2 (2024), 295–307.

[26] Jetty Kleijn, Maciej Koutny, and Łukasz Mikulski. 2020. Reaction Systems and Enabling Equivalence. *Fundamenta Informaticae* 171, 1-4 (2020), 261–277.

[27] Jetty Kleijn, Maciej Koutny, Łukasz Mikulski, and Grzegorz Rozenberg. 2018. Reaction systems, transition systems, and equivalences. In *Adventures between lower bounds and higher altitudes: essays dedicated to Juraj Hromkovič on the occasion of his 60th birthday*. Springer, 63–84.

[28] Jetty Kleijn, Maciej Koutny, and Grzegorz Rozenberg. 2011. *Modelling reaction systems with Petri nets.* Computing Science, Newcastle University.

[29] Maciej Koutny and Łukasz Mikulski. 2026. Encoding reaction systems in Petri nets. *Natural Computing* 25, 1 (2026), 8.

[30] Damian Kurpiewski, Wojciech Jamroga, Łukasz Maśko, Łukasz Mikulski, Witold Pazderski, Wojciech Penczek, and Teofil Sidoruk. 2022. Verification of Multi-Agent Properties in Electronic Voting: A Case Study. In *Proceedings of the 14th Conference on AiML (Artificial Intelligence and Mathematical Logic)*. College Publications, Rennes, France, 531–556. AiML 2022.

[31] Marta Kwiatkowska, Gethin Norman, and David Parker. 2011. PRISM 4.0: Verification of probabilistic real-time systems. In *International conference on computer aided verification*. Springer, 585–591.

[32] Kim G Larsen, Paul Pettersson, and Wang Yi. 1997. UPPAAL in a nutshell. *International journal on software tools for technology transfer* 1, 1 (1997), 134–152.

[33] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. 2009. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. In *Computer Aided Verification*. Springer Berlin Heidelberg, 682–688.

[34] Luca Manzoni, Antonio E Porreca, and Grzegorz Rozenberg. 2020. Facilitation in reaction systems. *Journal of Membrane Computing* 2, 3 (2020), 149–161.

[35] Artur Męski, Maciej Koutny, Łukasz Mikulski, and Wojciech Penczek. 2024. Reaction mining for reaction systems. *Natural Computing* 23, 2 (2024), 323–343.

[36] Artur Męski, Maciej Koutny, Łukasz Mikulski, Ion Petre, Wojciech Penczek, and Marcin Piatkowski. 2025. Model checking for distributed reaction systems with temporal-epistemic properties. *Natural Computing* 24, 1 (2025), 1101–1117.

[37] Artur Męski, Maciej Koutny, and Wojciech Penczek. 2019. Model checking for temporal-epistemic properties of distributed reaction systems. *School of computing technical report series* (2019).

[38] Artur Męski, Wojciech Penczek, and Grzegorz Rozenberg. 2015. Model checking temporal properties of reaction systems. *Information Sciences* 313 (2015), 22–42.

[39] Marco S Nobile, Antonio E Porreca, Simone Spolaor, Luca Manzoni, Paolo Cazzaniga, Giancarlo Mauri, and Daniela Besozzi. 2017. Efficient simulation of reaction systems on graphics processing units. *Fundamenta Informaticae* 154, 1-4 (2017), 307–321.

[40] Christos H Papadimitriou. 2003. Computational complexity. In *Encyclopedia of computer science*. 260–265.

[41] ReactICS 2025. https://reactics.org.

[42] ReactICS Video Demonstration 2025. https://youtu.be/GmIFmjh-F7g.

[43] Jorge Gómez Tejeda Zañudo, Maurizio Scaltriti, and Réka Albert. 2017. A network modeling approach to elucidate drug resistance mechanisms and predict combinatorial drug treatments in breast cancer. *Cancer convergence* 1, 1 (2017), 1–25.