UMR IRISA

Activity Report 2025

Team DEVINE

DEpendable distributed systems: formal VerificatIoN made Efficient

Joint team with Centre Inria de l'Université de Rennes

D4 – Language and Software Engineering

Université de Rennes | rennes | INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES | cnrs | Inria | Inserm | CentraleSupélec | IMT Atlantique Bretagne-Pays de la Loire École Mines-Télécom | Université Bretagne Sud ubs

Project-Team DEVINE

Creation of the Project-Team: 2024 January 01

Each year, Inria research teams publish an Activity Report presenting their work and results over the reporting period. These reports follow a common structure, with some optional sections depending on the specific team. They typically begin by outlining the overall objectives and research programme, including the main research themes, goals, and methodological approaches. They also describe the application domains targeted by the team, highlighting the scientific or societal contexts in which their work is situated. The reports then present the highlights of the year, covering major scientific achievements, software developments, or teaching contributions. When relevant, they include sections on software, platforms, and open data, detailing the tools developed and how they are shared. A substantial part is dedicated to new results, where scientific contributions are described in detail, often with subsections specifying participants and associated keywords. Finally, the Activity Report addresses funding, contracts, partnerships, and collaborations at various levels, from industrial agreements to international cooperations. It also covers dissemination and teaching activities, such as participation in scientific events, outreach, and supervision. The document concludes with a presentation of scientific production, including major publications and those produced during the year.

Keywords

Computer sciences and digital sciences

A2.5.5. – Software testing

A4.5. – Formal method for verification, reliability, certification

A4.5.2. – Model-checking

A8.9. – Performance evaluation

A8.11. – Game Theory

Other research topics and application domains

B5.1. – Factory of the future

B6.6. – Embedded systems

B7.1. – Traffic management

Contents

Project-Team DEVINE **1**

1 Team members, visitors, external collaborators **5**

2 Overall objectives **6**

3 Research program **7**
3.1 Efficient analysis of real-time systems . . . 7
3.2 Verification of distributed algorithms . . . 8
3.3 Optimization of multi-agent systems . . . 10

4 Application domains **11**

5 Social and environmental responsibility **11**
5.1 Footprint of research activities . . . 11
5.2 Impact of research results . . . 11

6 Highlights of the year **12**

7 Latest software developments, platforms, open data **12**
7.1 Latest software developments . . . 12
7.1.1 ParaGraphs . . . 12
7.1.2 SimGrid . . . 12
7.1.3 Ticynet . . . 13

8 New results **13**
8.1 New results on efficient analysis of real-time systems . . . 13
8.2 New results on verification of distributed algorithms . . . 15
8.3 New results on optimization of multi-agent systems . . . 16

9 Bilateral contracts and grants with industry **17**
9.1 Bilateral contracts with industry . . . 17

10 Partnerships and cooperations **17**
10.1 International initiatives . . . 17
10.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program . . . 17
10.1.2 Participation in other International Programs . . . 18
10.2 National initiatives . . . 18
10.2.1 ANR projects . . . 18
10.2.2 National Informal Collaborations . . . 19

11 Dissemination **20**
11.1 Promoting scientific activities . . . 20
11.1.1 Scientific events: selection . . . 20
11.1.2 Journal . . . 20
11.1.3 Leadership within the scientific community . . . 20
11.1.4 Research administration . . . 20
11.2 Teaching Supervision Juries Educational and pedagogical outreach . . . 21
11.2.1 Teaching . . . 21
11.2.2 Supervision . . . 21
11.2.3 Juries . . . 22
11.3 Popularization . . . 22
11.3.1 Participation in Live events . . . 22
11.3.2 Others science outreach relevant activities . . . 22

12 Scientific production **22**
12.1 Major publications 22
12.2 Publications of the year 23
12.3 Cited publications 25

1 Team members, visitors, external collaborators

Research Scientists

- Nathalie Bertrand [Team leader, INRIA, Senior Researcher, HDR]
- Loïc Hélouët [INRIA, Senior Researcher, HDR]
- Thierry Jéron [INRIA, Senior Researcher, HDR]

Faculty Members

- Loïc Germerie-Guizouarn [UNIV RENNES, Associate Professor]
- Aline Goeminne [ENS RENNES, Associate Professor, from Sep 2025]
- Julie Parreaux [UNIV RENNES, Associate Professor]

Post-Doctoral Fellows

- Sayan Mukherjee [INRIA then CNRS, Post-Doctoral Fellow, from Feb 2025]
- Gaëtan Staquet [INRIA, Post-Doctoral Fellow, until Aug 2025]

PhD Students

- Aymeric Come [INRIA]
- Victorien Desbois [NewLogUP, CIFRE]
- Pranav Ghorpade [UNIV. SYDNEY, located in Sydney]
- Luc Lapointe [ENS PARIS-SACLAY, located at LMF PARIS-SACLAY]
- Mathieu Laurent [ENS RENNES, shared with Magellan]
- Luca Paparazzo [UNIV RENNES]
- Antoine Thébault [Alstom , CIFRE, until Aug 2025]

Interns and Apprentices

- Maëlle Gautrin [ENS PARIS-SACLAY, Intern, from Mar 2025 until Aug 2025]
- Nowar Kazem [ENS RENNES, Intern, from May 2025 until Jul 2025]

Administrative Assistant

- Laurence Dinh [INRIA]

Visiting Scientist

- Srinivas Pinisetty [IIT BHUBANESWAR, from Jun 2025 until Jun 2025]

External Collaborator

- Ulrich Fahrenberg [EPITA then Université PARIS-SACLAY (since Sept. 2025)]

2 Overall objectives

Modern computer systems exploit concurrency in order to reach high levels of performance. Moreover, an increasing number of complex applications are also distributed, either to attempt to distribute the computation in search of performance gains, or because the system is by nature made of several components that communicate over a network. Both aspects bring difficulties in the development of dependable distributed applications. In fact, a large number of threads and components can mean that the set of possible behaviors of the system (thus, its set of configurations) is prohibitively large. Beyond software, many physical systems such as transportation networks or power grids are massively distributed by nature. There also, the variability of behaviors is extreme due to events interleaving and unexpected faults. In the absence of specific techniques for design and verification, ensuring the *functional correctness* of such systems (*i.e.* their ability to produce the expected outputs on given inputs) can become extremely difficult.

Apart from their distributed nature, many modern computers or physical systems rely on *extra-functional aspects*, such as real-time constraints and stochasticity. Expressing the correctness of many critical systems requires not only functional correctness, but also several more complex properties related to execution time, respect of deadlines, average performance, and probabilistic convergence. We will more generally call these aspects *quantitative* because they often involve enriching the description of the systems, their requirements, and their analysis with quantities such as time, probabilities, or other measures such as cost. As illustrative examples, we detail three applications in which a quantitative analysis is crucial:

- Time synchronization protocols are at the heart of highly distributed computing systems such as audio streaming over Ethernet, wireless sensor networks, accurate distance measurement through GPS satellites, and cloud-based applications. They must be fault-tolerant and should provide a time measure with high accuracy in order for these applications to be used, despite failures,latencies, etc.

- In distributed computing, randomization can yield more efficient solutions, or even permit solving problems that are otherwise unsolvable, such as the consensus problem in asynchronous message-passing systems in which as few as one process can crash.

- Statistics on unpredictable events are commonly used to enable performance evaluation, as in regulation for transportation networks where journey duration is represented as a random variable and potential failures as random events.

These three frameworks have in common the fact that purely functional verification techniques are certainly not sufficient for analyzing such applications. Yet reasoning about quantitative distributed systems is inherently difficult. For instance, the combination of distributed aspects and probabilities makes human reasoning difficult; quoting Lehmann and Rabin [63]: “proofs of correctness for probabilistic distributed systems are extremely slippery”. Real-time constraints bring additional difficulties since one not only must reason about the computations but also their timings which can induce particular interleavings between processes that can be difficult to predict and debug manually. Unfortunately, standard model-based verification techniques face scalability issues on the large-scale applications we target, such as transportation networks or time synchronization protocols in large networks.

Yet, when the considered distributed systems become bottlenecks in critical infrastructures, proving the correctness or assessing the performance of such systems in a reliable way is crucial. Indeed, failures can be prohibitive in terms of financial cost or even human loss. Safety-critical software or physical systems that operate safely and dependably yield a competitive advantage for industrials and reduce threats on the society. There is a need for the development of powerful algorithmic techniques for providing a rich set of guarantees on such distributed systems, by ensuring their functional correctness, while taking quantitative aspects into account. Our rationale is that *quantitative aspects must be fully integrated when reasoning about such systems, and must be at the heart of all phases of design, requirement development, testing, bug finding, formal verification and synthesis.* DEVINE aims to **develop algorithms for ensuring the dependability of quantitative distributed systems**. To achieve this objective, we will **develop model-based[1] scalable verification and optimization techniques**.

[1]We almost always target high-level models for distributed systems rather than the systems themselves. Bug finding in MPI (Message passing interface) programs —see Section 3.2— is an exception.

In order to ensure dependability of distributed systems with quantitative aspects, the research agenda of DEVINE is structured into the following axes.

1. Efficient analysis of real-time systems. The applicability of model-based formal methods to industrial-size real-time systems is challenged by the mix in models of discrete and continuous variables. Efficient model-checking, testing and runtime verification algorithms are needed to handle large models. We will also handle timing imprecisions and real-time security properties.

2. Verification of distributed algorithms. The behaviour of distributed algorithms and their implementations is hard to analyze due to asynchrony and failures. We will develop innovative bug-finding techniques for MPI programs and verification methodologies for pseudo-codes to prove their correctness independently of the number of processes.

3. Optimization of multi-agent systems. Standard optimization techniques do not scale to large multi-agent systems. We propose to formalize optimality, design efficient planning algorithms, and explore the trade-off between strategy optimality and computation cost.

A strength of the model-based techniques and tools we develop is that they are generic and high-level, so that they may prove useful in many application domains. The members of DEVINE aim at maintaining and increasing strong relations with industrial partners. As for software, on the one hand, we will develop prototype implementations to demonstrate the applicability of our techniques, and on the other hand, we will co-develop specific tools answering the needs of industrials.

3 Research program

3.1 Efficient analysis of real-time systems

Timed automata have been introduced in the early 1990s as a convenient framework for modelling and reasoning about real-time systems [27]. They combine discrete state space, to represent valuations of internal variables, and continuous variables called *clocks*, *e.g.* to measure delays between events. Timed automata and their variants have been extensively studied over the last 30 years, both on the theoretical and practical sides. Several efficient tools have been developed and applied to industrial case studies [31, 58]. The efficiency of these tools is however still challenged by the mix of discrete and continuous variables, which makes it hard to handle the state space symbolically. Our aim is to develop techniques, algorithms, and tools that scale to larger models, which would allow us to handle larger case studies.

Efficient model checking algorithms As in many applications of formal methods, formal verification suffers from state-space explosion, which limits the scalability of algorithms unless proper state-space reduction techniques are applied. In timed automata, state-space explosion can be caused by two factors: 1) a large discrete state space, *e.g.* if the system is composed of many subsystems, or contains several discrete variables; 2) a large number of clocks or complex timing constraints. While the first factor already appears in finite-state model checking, in timed automata, the state space can grow exponentially in the number of clocks requiring a particular care. State-of-the-art algorithms can efficiently deal with complex time constraints [54, 62, 65] but fail at analyzing models with both large discrete state spaces (for instance real-time distributed systems) and real-time constraints. This is however crucial to demonstrate the benefits of formal methods for real-time systems on realistic applications.

We will build novel tools based on compositional reasoning and predicate abstraction to achieve formal verification performance comparable with that of finite-state systems *without* explicit time constraints. In fact, although clock constraints do cause state-space explosion, they are often not the only source of complexity; so smart ways of handling them must be developed to handle large models. Ideally, one should be able to handle clock variables like any other variable in a program under verification. In this context, predicate abstraction [53, 69] is a promising direction since when used properly, clock variables can be treated as any other system variable; so several techniques from software model checking or finite-state automata can be applied. Compositionality is a well-known approach to handle larger models [41]; we will develop techniques to specifically handle the timing aspects in such approaches, and target the development of both fully automatized and interactive compositional model checkers. Last, some performance achievements

might appear by targeting specific applications and developing tailored algorithms, rather than relying on one generic algorithm. Our collaborations with industrial partners will guide us in this direction since these are opportunities to consider specific practical problems. In all these works, we will use and contribute to the open-source timed automata model checker TChecker [58].

Testing and runtime verification To extend the applicability of models like timed automata to verify industrial-size real-time systems, one can relax the exhaustiveness guarantee provided by model checking. Model-based test synthesis is one such technique (see, e.g. [73]): it consists in synthesizing, from a system model, sequences of actions to be performed on the implementation, in order to check that it behaves as specified. We will generate such test cases from real-time requirements, leveraging techniques recently developed by team members based on both test synthesis from game theory [57] and consistency checking [61]; this will complement the tool suite we developed in our collaboration with MERCE for checking consistency of real-time requirements, to obtain test cases for checking those requirements on real implementations.

Runtime monitoring [30] is another verification technique for assessing the validity of properties at runtime: it consists in observing the system as it executes and deciding as soon as possible whether the properties are satisfied or violated. In many contexts, the system is only partially observable, *i.e.* some internal actions are hidden to the monitor. Runtime monitoring is however limited to real-time systems that are not distributed. Our objective is to develop a framework for distributed runtime monitoring of real-time systems, in which several monitors observe components of the system, and exchange information so as to decide as early as possible on the validity of the property. Efficient solutions should limit the amount of communication as well as the computation time. Timed markings, a formalism we introduced and recently used to efficiently compute and manipulate sets of configurations [37], are a natural candidate tool to use. In the distributed setting, however, timed markings need to be reshaped to store sufficient information, and also to enable efficient updates with the observation and information stream.

Timing imprecisions The model of timed automata for real-time systems assumes arbitrary precision in time measurements. This artifact which is theoretically convenient has the drawback of missing behaviours if delays are slightly shifted. Since time drifts are inevitable in distributed systems, we will pursue the development of models, semantics and algorithms to take timing imprecisions into account for real-time systems. For instance, the efficiency of our recent algorithm for synthesizing *permissive strategies* [44] can be improved by relaxing its precision while keeping track of the amount of approximation in the computation.

Timing imprecisions are also very relevant in *online* techniques, such as monitoring, testing and learning, since those techniques involve interactions with physical implementations. Because of those imprecisions, the observation of the system may be inexact, and the actions performed on the system may be slightly shifted, so that a given sequence of inputs may result in different outputs. This does not fit with our current techniques [56, 57], and we will have to develop specific approaches to take such imprecisions into account.

Real-time security properties Most often in formal methods properties are defined at the level of individual behaviours: for instance, an execution of a program is terminating, or it isn't. However, in order to express security properties such as non-interference, one needs to reason on pairs of executions (for instance to guarantee that observing the control flow of a program does not leak information on a private key), or more generally sets of executions. So-called *hyperproperties*, introduced a decade ago [43, 42], allow one to compare executions of untimed models. Dealing with real-time in that context is a great challenge since one needs to compare dates of event occurrences. So far it only has been considered in a discrete-time setting [36]. We will provide verification algorithms in the continuous-time case, with the objective of addressing security properties related to timing issues, such as covert communication induced by timing channels.

3.2 Verification of distributed algorithms

Distributed algorithms are central to many domains such as scientific computing, telecommunications and the blockchain. Even when they aim at performing simple tasks, their behaviour is hard to analyze, due mainly to the asynchrony between the processes and to the presence of faults (crashes, message losses, etc.). We aim on the one hand at designing efficient techniques for verification of HPC (High Performance Computing)

programs, and on the other hand at establishing correctness of distributed algorithms independently of the number of participants.

Efficient bug finding for MPI program HPC applications in the MPI (message passing interface) programming model consist of distributed programs that communicate asynchronously through FIFO (First In First Out) channels. Finding bugs, or proving their absence, in such programs is challenging because of the complexity due to concurrency and communication. The goal of the McSimGrid tool [66] is to automatically check properties directly on the programs, considering all alternative executions of the program for a fixed input. Rather than proving correctness, it aims at finding concurrency and communication bugs efficiently. As often in verification, scaling to real programs requires techniques that avoid the state-space explosion. One way to do so, is to use dynamic partial order reduction (DPOR) [52, 29, 68], which cleverly exploits the independence of concurrent events to reduce the state space to be explored. Beyond plain DPOR algorithms, in order to go an order of magnitude further in terms of program size, we propose to combine DPOR with other efficient bug-finding techniques, such as directed model checking [50]. Directed model checking prioritizes the state-space exploration using A*-like algorithms (that is optimal pathfinding algorithms), relying on approximate distances to a goal. In the context of MPI programs, the definition of appropriate distances is crucial for balancing the trade-off between precision and computation cost, that impact the time to find an error in two opposite ways. Alternatively, one can combine DPOR with other bug-finding techniques, by bounding some values [45] or progressively refining the independence relation. In order to evaluate these heuristics for MPI programs, the various approaches will be implemented in McSimGrid and benchmarked against academic case studies.

Parameterized verification of pseudo-codes The correctness of distributed algorithms should be established independently of actual setup, *i.e.* the number of processes, the potential failures, and the communication topology when relevant. Parameterized verification answers this need by handling multiple model-checking queries at once; it also often comes with cutoff results that provide bounds on the parameters for which a bug can happen, in case the correctness does not hold. To overcome the general undecidability of the verification of properties for distributed systems with an unbounded number of processes [28], we propose two natural approaches: on the one hand exhibiting models for specific distributed algorithms with decidable parameterized verification, and on the other hand developing incomplete verification algorithms that may not terminate or may be inconclusive on some instances. We illustrate these alternatives on the two archetypal frameworks below, on which we will focus first.

Parameterized verification of models of MPI programs. Complementary to the objective of efficiently finding bugs in MPI programs, one can come up with models of MPI programs and study their parameterized verification problem. A relevant way to classify MPI programs with respect to model checking is by the communication primitives they use and possibly the communication topologies: mailboxes are typically represented as queues or bags, for instance. Our objective is to define classes of models for MPI programs with decidable parameterized verification. Most likely these models will approximate the actual behaviour of the programs. A prototype implementing parameterized verification for MPI programs would be a major breakthrough compared to the fixed-instances existing tools such as CIVL [64].

Parameterized verification of randomized distributed algorithms. Randomization is an elegant tool to design efficient algorithms or even to solve problems otherwise unsolvable, especially in distributed computing, where probabilities break symmetry between the components. Till now, automated proofs of randomized distributed algorithms remain limited to restricted types of algorithms, restricted classes of schedulers, and restricted properties [32, 33, 34]. Leveraging parameterized verification techniques to handle randomized distributed algorithms raises the challenge that performance typically depend on the number of participants, whereas standard parameterized verification techniques abstract away this parameter. *Counter-example guided abstraction refinement* (CEGAR) approaches have proven extremely efficient on non-randomized fault-tolerant distributed algorithms [35, 70]. Building on the latter, and on an existing CEGAR framework for finite-state probabilistic systems [59, 49], we aim at targetting randomized distributed algorithms, which is non-trivial: appropriate predicates must be defined, and counterexamples must be generic enough to account for sets of parameter values.

3.3 Optimization of multi-agent systems

While verification merely checks that a property holds, control aims at optimizing the system performance related to quantities such as time or energy consumption. A controller resolves choices that are left open, typically during early design phases to adapt to a particular context. Resolving choices amounts to selecting a strategy, in order to optimize certain quantitative objectives. When choices are left to several agents, their interaction is represented by a game. Solving a game amounts to computing optimal strategies for each of the agents. Controlling large systems or solving large games is even more challenging than verification: the distributed nature of multi-agent systems leads to exponential blowup of the state space. Admittedly, standard optimization techniques [67] such as value iteration or policy iteration (that is, iterative fixpoint computations of optimal strategies) do not scale to large state spaces. Motivated by traffic management in transportation systems, we will attack this problem by formalising optimality logically, designing efficient planning algorithms or computing close-to-optimal strategies with guarantees.

Formalising optimality in quantitative games Before even computing optimal or quasi-optimal strategies for multi-agent systems, one needs to formally define these. Strategy Logic (SL for short) is a temporal logic for expressing complex properties of multiple-player games [40]. It can be used to express, *e.g.* the existence of equilibria, as well as properties of the outcomes of those equilibria. However, this logic is not suited to games with quantitative aspects: it cannot handle quantitative (as opposed to Boolean) payoffs for the players, which are needed for fine-grained representation of their preferences. We will continue the exploration of such quantitative features of SL, building on our recent works [38] on *fuzzy* SL (which was, to our knowledge, the first decidable quantitative extension of SL). Due to the high complexity of model checking for SL, we will also consider fragments of the logic, targeting good trade-offs between expressiveness and complexity.

Computing sub-optimal strategies with guarantees In non-critical contexts, sub-optimal strategies can be sufficient, and their computation can be less costly than optimal ones. In game theory, Simon [71] introduced the portmanteau word *satisficing* that combines *satisfying* and *sufficient*, to describe situations in non-cooperative games, where agents may opt for a non-optimal strategy if they think their reward is good enough and that the effort (be it time, energy, or computational power) needed to get closer to the optimal reward would be prohibitive. This notion has been addressed in economy as an alternative to optimality in standard reinforcement learning (RL) algorithms (see, e.g. [72]). Strategies in RL are built to optimize quantitative goals but allow stopping construction when side measures such as *regrets* or *risk* exceed a fixed aspiration level. Building on our expertise in various aspects of quantitative games [39, 55], we aim at computing good-enough strategies by relating the improvement, *e.g.* measured as a reward, that can be obtained by changing a strategy and the extra effort needed to play the new strategy.

We will also address the scalability issue in optimal control problems in Markov decision processes by targeting distributed systems. This direction will combine approaches both from the formal verification world, such as abstraction techniques, and bounded verification [47], and from the reinforcement learning world, such as multi-agent reinforcement learning techniques and approximate solution methods (e.g. deep learning). This rich set of techniques will allow us to solve applications such as train regulation problems.

Application to traffic management in transports Urban rail transportation calls for optimization techniques in order to improve the users experience in terms of metro punctuality, regularity, and time needed to resume to a nominal behaviour after an incident, to name a few. For a single metro line, efficient traffic management techniques exist and mainly consist in adapting fleets sizes, speeds of trains and dwell times [46]. However, in large cities, the public transport demand is expected to grow tremendously (49% larger in 2050 than in 2012) with huge economic and environmental impacts. Improving efficiency of public transports must include several transport modes and several operators, making the state-of-the-art techniques unapplicable. Multi-modal transport raises the issue that operators, clients, and decision-makers have their own objective. Optimizing traffic management in this context is a challenging task: it amounts to solving a huge multi-player game with multiple objectives. Our aim is to compute good-enough strategies for each actor in these games. Despite the large size of transport networks, recently developed tools for concurrent hybrid models enable numerical methods, and can be used to learn regulation mechanisms [51]. Symbolic representations of sets of states and distributions [60] is also promising to speed up simulation, discover appropriate abstractions and therefore enable efficient traffic management.

4 Application domains

A strength of the model-based techniques and tools we develop is to be generic and high-level so that they may find applications in many domains. Members of the team already have long-lasting collaborations with industrial partners in transportations systems and factory automation, and new collaborations in blockchain technologies are emerging. Our experience demonstrates that new applications often feed our research with new and challenging problems. We are however aware of the time required to invest into a new domain. Our strategy is hence to be opportunistic and remain open to any potential application of our techniques, within the limits imposed by the size of the team.

Industrial transfer In terms of industrial technology transfer, we will aim in DEVINE at maintaining and increasing strong relations with industrial partners from various application domains. The genericity of formal methods and verification techniques indeed open many opportunities for industrial transfer. We aim at transferring knowledge to researchers and engineers in industrial partners through CIFRE PhD projects, and impact the products of our industrial partners with innovative techniques that can lead to patents and be used in production.

These objectives are very sensible given the long-term experience of several members of DEVINE in collaborations with Mitsubishi Electric (MERCE) and Alstom Transport. These have led to several CIFRE PhDs, to training of an engineer from MERCE, to filing of patents, publications, and to the transfer of a software tool (SIMSTORS) to Alstom Transport.

Transfer to other CS fields Apart from transfer to industries, we will also aim at impacting fields in computer science other than formal methods. The two fields we target are distributed computing and scientific computing, which would in our opinion benefit from our expected contributions in formal verification of distributed algorithms and verification of MPI programs. We plan to continue and strengthen our impact to these fields, mainly through publications in conferences and journals of the respective domains, that members of DEVINE started to do via collaborations with distributed computing and MPI experts.

5 Social and environmental responsibility

5.1 Footprint of research activities

Some members of the team individually take part in the TCS4F (Theoretical Computer Science for Future) initiatives.

Since COVID, the carbon footprint of travels related to our research activities significantly decreased. Some members of the team no longer fly to attend conferences, and everyone carefully chooses the conferences where they submit not only in terms of reputation, but also taking into account the location or the possibility to attend online.

5.2 Impact of research results

DEVINE team members maintain long-term relationships with industrials. These tight collaborations could have impact in the future, as illustrated below by two examples.

- Since 2022, DEVINE collaborates with Alstom Transport on improvement of traffic management in metro systems (CIFRE grant of Antoine Thébault). One of the concerns in this study is to reduce the energy consumed by metros using smart controllers. Though these studies are currently conducted at a theoretical level and tested in silico, their transfer to running systems may help reducing the energy used in urban transports in the future.

- In 2025, DEVINE and other Inria teams started a *Défi commun* with MERCE (Mitsubishi Electric Research Center Europe). The outcomes of this 4-year long project should impact the design of safe systems in particular in several domains like railway, automotive, FA, elevators, where Mitsubishi is a world-class leader.

6 Highlights of the year

MERCE (Mitsubishi Electric Research Center Europe) and Inria have launched the *Défi commun* FRAIME mid-september 2025. In FRAIME, we propose to explore on the one hand how Formal Methods can provide guarantees on AI systems, and on the other hand how AI can help Formal Methods to be more efficient and easier to use by practitioners. The vision is to intertwine Formal Methods and AI to efficiently design safe systems.

7 Latest software developments, platforms, open data

7.1 Latest software developments

7.1.1 ParaGraphs

Name: ParaGraphs

Keywords: Game theory, Symbolic verification

Functional Description: ParaGraphs implements two new algorithms based on antichains, and the original PSPACE algorithm for solving parameterized concurrent games with reachability objectives. It is an open-source C++20 tool. It currently supports constraints given as finite unions of intervals, but is designed in a modular way and permits to easily define new types (for instance semi-linear sets), as well as specific implementations of the algorithms for these types.

Release Contributions: First version of the software.

URL: https://gitlab.inria.fr/gstaquet/paragraphs

Contact: Nathalie Bertrand

Participants: Gaetan Staquet, Nathalie Bertrand

7.1.2 SimGrid

Keywords: Large-scale Emulators, Grid Computing, Distributed Applications

Scientific Description: SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The simulation engine uses algorithmic and implementation techniques toward the fast simulation of large systems on a single machine. The models are theoretically grounded and experimentally validated. The results are reproducible, enabling better scientific practices.

Its models of networks, cpus and disks are adapted to (Data)Grids, P2P, Clouds, Fogs, Clusters and HPC, allowing multi-domain studies. It can be used either to simulate algorithms and prototypes of applications, or to emulate real MPI applications through the virtualization of their communication, or to formally assess algorithms and applications that can run in the framework.

The formal verification module explores all possible message interleavings in the application, searching for states violating the provided properties. This tool can be used to assess safety properties over arbitrary and legacy codes, thanks to a system-level introspection tool that provides a finely detailed view of the running application to the model checker. This can for example be leveraged to verify arbitrary MPI code written in C/C++/Fortran.

Functional Description: SimGrid is a simulation toolkit that provides core functionalities for the simulation of distributed applications in large scale heterogeneous distributed environments.

Release Contributions: Breaking the seal: v4.0 was not the final release.

* Allow one to unseal netzones to modify the platform even after the simulation start. * The model-checker can now report memory race conditions (see tutorial). * Pip builds should now work out of the box. * (+ the usual bug fixes overall, and improvements to the Java/Python bindings).

News of the Year: Most of the scientific work in SimGrid in 2025 occurred in the embedded model checker. Our work could profit of the 2 major releases of SimGrid in 2025 (v4.0 in March and Release v4.1 in November). We first worked on improving the performance of the verification process, and fixed some bugs. Our preferred algorithm (ODPOR reduction + Best-first exploration to enable random walk) is now much faster and consumes much less memory, being 5x faster on small scenarios. But since performance is now linear with the number of states, while it used to be polynomial, 5x faster is the lowest performance boost you can expect from the new version. The number of states to explore for a given scenario is still the same (ODPOR was not improved), but we now can explore these states much faster.

In addition, we added the ability to check for race conditions in the user code. This feature relies on a specific pass to the clang LLVM compiler, to instrument the memory accesses. We use this feature for teaching purposes at our institutions.

We are working on a parallel explorer leveraging all cores to accelerate the exploration, but unfortunately, we did not manage to find all the bugs in our parallel explorer yet. We are working on verifying SimGrid with itself to hammer this bug out.

URL: https://simgrid.org/

Publication: hal-04909441

Contact: Martin Quinson

Participants: Mathieu Laurent, Anne-Cécile Orgerie, Arnaud Legrand, Augustin Degomme, Arnaud Giersch, Frédéric Suter, Martin Quinson, Samuel Thibault

Partners: CNRS, ENS Rennes

7.1.3 Ticynet

Name: Time Cyclic Networks Tool

Keywords: Simulator, Public transport, Transport model

Functional Description: Ticynet is a tool for the analysis of quantitative properties of urban transport networks and for the optimization of these properties through control techniques.

Specifically, Ticynet allows users to design a metro network and analyze the energy savings achieved through an appropriate fleet management strategy.

In its current version, Ticynet allows to:

- model an urban transport line (tracks and vehicle fleet) - model the regenerative braking phenomenon - integrate a controller managing vehicle departures, arrivals, and speeds - simulate the complete system to quantify the energy saved ratio.

Release Contributions: Initial version, demonstrator for the ANR Project BisoUS.

Contact: Loic Helouet

8 New results

This section presents the contributions of the team members in 2025.

8.1 New results on efficient analysis of real-time systems

Participants: Ulrich Fahrenberg, Loïc Germerie-Guizouarn, Loïc Hélouët, Thierry Jéron, Sayan Mukherjee, Julie Parreaux, Ocan Sankur, Gaëtan Staquet.

On this research axis, our contributions this year concern: runtime enforcement, model learning for real-time systems, timed games, and compositionality of transducers.

Runtime enforcement In the paper [14], we deal with the problem of runtime enforcement (RE) in the context of reactive systems, which consists in modifying the outputs of a system minimally to ensure its correctness. In contrast to enforcers that can postpone events via buffering, enforcers for reactive systems must operate within the same reactive cycle, always yielding a (possibly modied) output. For safety properties, an enforcer makes sure to satisfy the property at each step. However, for general regular properties, one can only expect to satisfy the property eventually. There is then a risk that even under enforcement the satisfaction is indefinitely delayed, and the property is never actually satisfied. Forcing the satisfaction of regular or ω-regular properties has been considered using bounded fairness and prompt eventuality. In this paper, we propose a new runtime enforcement framework for regular properties with prompt eventualities. Given an automaton specifying a property φ and a bound k, the enforcer should never falsify φ more than k consecutive steps. We formally define this RE problem, characterize k-enforceability of automata, and exhibit the construction of the enforcer. Rather than fixing k, we also study whether k can be computed to ensure k-enforceability or some maximal coverage of φ. We implement the k-prompt enforcement framework, and demonstrate its behaviour with varying k.

Learning models for real-time systems In [22] we present a state-merging algorithm for learning timed languages definable by Event-Recording Automata (ERA) using positive and negative samples in the form of symbolic timed words. Our algorithm, (Learning Event-recording Automata Passively), constructs a possibly nondeterministic ERA from such samples based on merging techniques. We prove that determining whether two ERA states can be merged while preserving sample consistency is an NP-complete problem, and address this with a practical SMT-based solution. Our implementation demonstrates the algorithm's effectiveness through examples. We also show that every ERA-definable language can be inferred using our algorithm with a suitable sample.

In [18] we present the first algorithm for query learning Mealy machines with timers in a black-box context. Our algorithm is an extension of the $L^\sharp$ algorithm of Vaandrager et al. to a timed setting. We rely on symbolic queries which empower us to reason on untimed executions while learning. Similarly to the algorithm for learning timed automata of Waga, these symbolic queries can be realized using finitely many concrete queries. Experiments with a prototype implementation show that our algorithm is able to efficiently learn realistic benchmarks.

Timed Games Weighted Timed Games (WTG for short) are the most widely used model to describe controller synthesis problems involving real-time issues. We consider optimal reachability objectives, in which one of the players, that we call Min, wants to reach a target location while minimising the cumulated weight. Unfortunately, WTGs are notoriously difficult, and undecidable with two or more clocks. As a consequence, one-clock WTGs have attracted a lot of attention, especially because they are known to be decidable when only non-negative weights are allowed. However, when arbitrary weights are considered, despite several recent works, their decidability status was still unknown. In [9], we solve this problem positively and show that the value function can be computed in exponential time (if weights are encoded in unary). Alternatively to restricting to a single clock, several conditions, one of them being divergence, have been given to recover decidability. In such weighted timed games (like in untimed weighted games in the presence of negative weights), Min may need finite memory to play (close to) optimally. This is thus tempting to try to emulate this finite memory with other strategic capabilities. In [10], we allow the players to use stochastic decisions, both in the choice of transitions and of timing delays. We give a definition of the expected value in weighted timed games. We then show that, in divergent weighted timed games as well as in (untimed) weighted games (that we call shortest-path games in the following), the stochastic value is indeed

equal to the classical (deterministic) value, thus proving that Min can guarantee the same value while only using stochastic choices, and no memory.

Compositionality of transducers Deterministic two-way transducers with pebbles (aka pebble transducers) capture the class of polyregular functions, which extend the string-to-string regular functions allowing polynomial growth instead of linear growth. They are the object of [20]. One of the most fundamental operations on functions is composition, and (poly)regular functions can be realized as a composition of several simpler functions. In general, composition of deterministic two-way transducers incur a doubly exponential blow-up in the size of the inputs. A major improvement in this direction comes from the fundamental result of Dartois et al. [48] showing a polynomial construction for the composition of reversible two-way transducers. A precise complexity analysis for existing composition techniques of pebble transducers is missing, but they rely on the classic composition of two-way transducers and inherit the double exponential complexity. To overcome this problem, we introduce in this paper reversible pebble transducers. Our main results are efficient uniformization techniques for non-deterministic pebble transducers to reversible ones and efficient composition for reversible pebble transducers. Our objective is now to lift these result to the timed setting.

8.2 New results on verification of distributed algorithms

Participants: Nathalie Bertrand, Pranav Ghorpade, Thierry Jéron, Mathieu Laurent.

On this research axis, our contributions this year range from fundamental to more applied: true concurrency models such as Petri nets, dynamical partial order reduction techniques, and verification of blockchain protocols.

True concurrency models In [13] we consider a new approach for the concurrent semantics of Petri nets. Petri nets and their variants are often considered through their interleaved semantics, i.e. considering executions where, at each step, a single transition fires. This is clearly a miss, as Petri nets are a true concurrency model. This paper revisits the semantics of Petri nets as higher-dimensional automata (HDAs) as introduced by van Glabbeek, which methodically take concurrency into account. We extend the translation to include some common features. We consider nets with inhibitor arcs, under both concurrent semantics used in the literature, and generalized self-modifying nets. Finally, we present a tool that implements our translations.

Dynamic Partial Order Reduction Assessing the correctness of distributed and parallel applications is notoriously difficult due to the complexity of the concurrent behaviors and the difficulty to reproduce bugs. In this context, Dynamic Partial Order Reduction (DPOR) techniques have proved successful in exploiting concurrency to verify applications without exploring all their behaviors. However, they may lack of efficiency when tracking non-systematic bugs of real size applications. In this work [21], we suggest two adaptations of the Optimal Dynamic Partial Order Reduction (ODPOR) algorithm with a particular focus on bug finding and explanation. The first adaptation is an out-of-order version called RFS ODPOR which avoids being stuck in uninteresting large parts of the state space. Once a bug is found, the second adaptation takes advantage of ODPOR principles to efficiently find the origins of the bug.

Verification of blockchain protocols Blockchains use consensus protocols to reach agreement, e.g., on the ordering of transactions. DAG-based consensus protocols are increasingly adopted by blockchain companies to reduce energy consumption and enhance security. These protocols collaboratively construct a partial order of blocks (DAG construction) and produce a linear sequence of blocks (DAG ordering). Given the strategic significance of blockchains, formal proofs of the correctness of key components such as consensus protocols are essential. The contribution [16] presents safety-verified specifications for five DAG-based consensus protocols. Four of these protocols—DAG-Rider, Cordial Miners, Hashgraph, and Eventual Synchronous BullShark—are well-established in the literature. The fifth protocol is a minor variation of Aleph, another well-established protocol. Our framework enables proof reuse, reducing proof efforts by almost half. It

achieves this by providing various independent, formally verified, specifications of DAG construction and ordering variations, which can be combined to express all five protocols. We employ TLA+ for specifying the protocols and writing their proofs, and the TLAPS proof system to automatically check the proofs. Each TLA+ specification is relatively compact, and TLAPS efficiently verifies hundreds to thousands of obligations within minutes. The significance of our work is two-fold: first, it supports the adoption of DAG-based systems by providing robust safety assurances; second, it illustrates that DAG-based consensus protocols are amenable to practical, reusable, and compositional formal methods.

8.3 New results on optimization of multi-agent systems

Participants: Nathalie Bertrand, Aymeric Come, Loïc Hélouët, Luca Paparazzo.

On this research axis, our contributions this year concern the modelling and optimization of transportation networks, and the optimization of probabilistic systems.

Modelling and optimizing transportation networks With the application of optimization of urban transportation systems in mind, we proposed several models for multi-agent systems and studied relevant verification questions.
In [23] we consider a timed model tailored to study energy savings in transport networks. It focuses on regenerative braking, a situation where the kinetic energy of a vehicle is transformed in electrical energy and sent back in the electrical network. We first describe transport networks, and formalize their semantics as timed runs of an equivalent network of timed automata (NTA). We then consider three problems. The transfer existence problem checks whether a network has the ability to save energy. The ratio maximization problem aims at computing the maximal ratio of energy saved by time unit in the long run, and the threshold problem consists in verifying the existence of a strategy allowing the saving of more energy than a fixed minima. We show that the transfer problem is a reachability question in the region automaton of the NTA, which can be solved in PSPACE. The ratio problem can be solved using Karp's algorithm on a corner point abstraction for the NTA, yielding an EXPTIME complexity. Finally, the threshold problem requires to address properties of elementary cycles and finite paths of the corner-point automaton, yielding a PSPACE complexity.
In [17] we consider a new model for multi-agent systems. This contribution introduces collaborative reachability games with energy constraints. In the considered arenas, agents can spend or gain energy during moves, or share it with their peers if their current position allows it. We study several variants of energy reachability games where agents move either synchronously or asynchronously, and with/without constraints on energy transfers among peers. We show that these problems have different complexities ranging from NP to EXPSPACE.

Optimizating of probabilistic systems Markov chains and Markov decision processes (MDPs) are well-established probabilistic models. We proposed two complementary contributions to the analysis of MDPs: on the one hand an innovative depth-first search strategy, and on the other hand approximation algorithms for large of infinite-state models.

In the paper[8] we consider new algorithms to compute fixpoints in Markov Decison Processes. Value and policy iteration are classical algorithms to maximize the average discounted reward of an MDP. They rely on a breadth-first exploration strategy in the future of each state to update its value and possibly change the action policy at this state. This paper revisits this paradigm and examines a depth-first search strategy. It reformulates the average reward computation as an integral over (future) paths that is better expressed in the formalism of weighted automata. Policy evaluation can then be solved by a Floyd-Warshall algorithm, which gathers at once the rewards along possibly infinite runs. This reformulation opens the way to new approximation schemes for the value function. We show that the same approach also gives access to other quantities of interest, as the gradient of the average reward with respect to model or policy parameters, or the variance of the reward. The behaviors and performances of this value estimation scheme are illustrated on several benchmarks.

While finite Markov models are well-understood, analyzing their infinite counterparts remains a significant challenge. Decisiveness has proven to be an elegant property for countable Markov chains: it is general

enough to be satisfied by several natural classes of countable Markov chains, and it is a sufficient condition for simple qualitative and approximate quantitative model-checking algorithms to exist. In contrast, existing works on the formal analysis of countable MDPs usually rely on ad hoc techniques tailored to specific classes. We provide in [25] a general framework to analyze countable MDPs by extending the notion of decisiveness. Compared to Markov chains, MDPs exhibit extra non-determinism that can be resolved in an adversarial or cooperative way, leading to multiple natural notions of decisiveness. We show that these notions enable the approximation of reachability and safety probabilities in countable MDPs using simple model-checking procedures. We then instantiate our generic approach to two concrete classes of models inducing countable MDPs: non-deterministic probabilistic lossy channel systems and partially observable MDPs. This leads to an algorithm to approximately compute safety probabilities in each of these classes.

9 Bilateral contracts and grants with industry

9.1 Bilateral contracts with industry

Collaboration with Alstom Transport - P22

Participants: Loïc Hélouët, , Antoine Thébault.

DEVINE is involved in a long-term collaboration with Alstom Transport on the topic of urbain train systems regulation. Alstom and DEVINE jointly supervised a CIFRE PhD on the topic of smart traffic management (PhD of Antoine Thébault, ANRT grant 2022-0444, 2022-2025). This PhD addressed the two following objectives: optimize traffic management using concurrent models on one hand, and learning techniques (neural networks training, decision tree synthesis) on the other hand to synthesize controllers.

10 Partnerships and cooperations

10.1 International initiatives

10.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program

SINCRET

Participants: Loïc Germerie-Guizouarn, Thierry Jéron, Sayan Mukerjee.

Title: DST/Inria Associate Team SINCRET, Scalable and INCREmental security monitoring and enforcement for Timed systems

Website: link to Website

Partner Institution: IIT Bhubaneswar, India

Led by: Thierry Jéron (Inria) and Srinivas Pinisetty (IIT Bhubaneswar)

Objectives: The objective of this project is to study the enforcement of timed properties for cyber physical systems (CPS) in a synchronous context, and study their incremental composition. We first consider an enforcement monitor synthesis framework for reactive CPS focused on discrete timed properties. This year we proposed a serial composition scheme, delineate the sub-class of properties that are enforceable, and develop a serial composition scheme. A prototype is developped by our partner and experimented on a case study of a swarm of drones. A publication is under review for a journal. We also worked on Prompt Runtime Enforcement in the untimed setting, and got a paper published at ATVA 2025 [14]. We currently continue this work in the timed setting.

10.1.2 Participation in other International Programs

IRL Relax

Participants: Nathalie Bertrand, Loïc Hélouët.

Several members of DEVINE have tight relationships with colleagues at Chennai Mathematical Institute and IIT Bombay. For this reason, we are involved in the CNRS IRL Relax, an Indo-French joint research unit dedicated to research in theoretical computer science and in mathematics, their interactions and their applications.

10.2 National initiatives

10.2.1 ANR projects

ANR MAVeriQ: Methods of Analysis for Verification of Quantitative properties (2021-2025)

Participants: Aymeric Côme, Loïc Hélouët, Nathalie Bertrand.

Website: https://www.irif.fr/users/maveriq/index

Led by Aldric Degorre (IRIF); Local coordinator Éric Fabre.

Partners: IRIF, LMF, Inria Rennes/IRISA, LACL, Verimag.

Objectives: The objective of this project is to develop unified frameworks for quantitative verification of timed, hybrid, and stochastic systems. We believe such a unification is possible because common patterns are used in many cases. The project targets in particular: • systematization of quantitative properties and their use cases • substantial progress in the algorithms of quantitative verification; • practical methodology for stating and verifying quantitative properties of systems. The aim of MAVeriQ is to progress towards this unification, by gathering skills on timed and stochastic systems and on quantitative verification under a common roof, to jointly address open challenges in quantitative model-checking and quantitative validation. One such challenge we will address is robustness of quantitative models, that is, resilience to small perturbations, which is crucial for implementability. Unified methods developed in the project (such as robustness analysis and simulation techniques) will be showcased in different case studies in the domain of CPS (in particular automotive control), showing that such a system can be verified in different ways without leaving this framework.

ANR BisoUS: Better Synthesis for Underspecified Quantitative Systems (2023-2027)

Participants: Nathalie Bertrand, Loïc Hélouët, Nicolas Markey, Julie Parreaux, Ocan Sankur.

Website: https://anr-bisous.ls2n.fr

Led by Didier Lime (LS2N); Local coordinator Nathalie Bertrand.

Partners: LS2N, Inria Rennes/IRISA, LIPN, LMF.

Objectives: When designing complex and critical systems (planes, autonomous vehicles, etc.), it is crucial to be able to give guarantees that the system works as intended, which is often done through comprehensive testing. The goal of project BisoUS is to provide stronger guarantees, based on formal methods, and to detect problems as early as possible: solving them is then easier and cheaper. Unfortunately this is a hard problem because some design choices may not have been done yet, and some key features (e.g.

speed of a CPU) are then not known precisely enough. In project BisoUS we develop formal methods, based on model-checking and synthesis to work with expressive modelling formalisms encompassing parameters, cost/rewards, and games on graphs to meet those challenges.

ANR PaVeDyS: Parametric Verification of Dynamic Distributed Systems (2024-2027)

Participants: Nathalie Bertrand, Loïc Germerie-Guizouarn, Thierry Jéron, Ocan Sankur.

Website: https://raduiosif.github.io/PAVEDYS/

Led by Radu Iosif (Verimag); Local coordinator Nathalie Bertrand.

Partners: Verimag, Inria Rennes, IRIF, LaBRI.

Objectives: Applications of distributed systems are omnipresent. They allow sharing resources and data. They are used to coordinate activities across multiple nodes, as in geographically distributed systems. Furthermore, they increase the resilience of systems through fault tolerance, availability, and recovery mechanisms. Designing, understanding, and validating distributed systems are challenging because of the huge number of interactions between components, some potentially leading to unpredictable scenarios. Early detection of design errors is not only crucial for financial reasons, but it is often the only feasible way to find critical errors. The methods for ensuring the correctness of distributed systems are not yet mature. This is particularly the case for the mechanized reasoning methods that we propose to develop in this project.

Défi FRAIME (2025-2029)

Participants: Nathalie Bertrand, Ocan Sankur.

Led by Nathalie Bertrand (DEVINE) and David Mentré (MERCE).

Partners: Gallinette, EPICURE, ARGO, DiverSE, DEVINE, MERCE.

Objectives: Together with MERCE (Mitsubishi Electric Research Center Europe), Inria launched mid september 2025 a joint Défi. FRAIME gathers five Inria teams (DEVINE, Epicure, Gallinette, DiverSE, ARGO) and two research teams from MERCE Digital Information Systems division (Information and Network Systems and Synergistic Autonomous Systems). In FRAIME, we propose to explore on the one hand how Formal Methods can provide guarantees on AI systems, and on the other hand how AI can help Formal Methods to be more efficient and easier to use by practitioners. The vision is to intertwine Formal Methods and AI to efficiently design safe systems. The project is driven by industrial use cases in the expertise of MERCE and targets the following four challenging objectives: user-assisted factory automation code generation, correctness of C programs, safe and optimal system control, easier and more automated proof assistant.

10.2.2 National Informal Collaborations

The team collaborates with the following researchers:

- Patricia Bouyer (LMF, ENS Paris-Saclay) on quantitative aspects of verification and game models for parameterized systems;
- Luc Dartois (FEMTO-ST, Université de Besançon) and Paul Gastin (LMF, ENS Paris-Saclay) on transducer models and verification of transformations;

11 Dissemination

11.1 Promoting scientific activities

11.1.1 Scientific events: selection

Chair of conference program committees

- Nathalie Bertrand serves as PC chair for the international conference FoSSaCS'26. This activity spans from summer 2025 (starting with identifying PC members) to spring 2026 (ending with the conference being held).

Member of the conference program committees

- Nathalie Bertrand has served on the PC of the international conferences FoSSaCS'25 and MFCS'25.
- Loïc Hélouët has served as PC member of the international conferences Petri Nets 2025, NETYS 2025, and ATMOS 2025.
- Thierry Jéron was in the program committee of SAC-SVT'25.
- Julie Parreaux was in the program comittee of ICFEM'25.

Reviewer
DEVINE members regularly write reviews for the central international conferences of our field: LICS, ICALP, CAV, Concur, FSTTCS, etc.

11.1.2 Journal

Member of the editorial boards
Nathalie Bertrand is an editorial board member for Journal of Logical and Algebraic Methods in Programming (JLAMP) and for Theoretical Computer Science (TCS).

Reviewer reviewing activities
All members of the team reviewed a number of papers for international journals in their field of expertise.

11.1.3 Leadership within the scientific community

Nathalie Bertrand is the co-head of the French working group on verification of GDR-IFM: GT Vérif.

11.1.4 Research administration

- Nathalie Bertrand is member of the *Formation Spécialisée de Site* at Inria center of the University of Rennes.
- Nathalie Bertrand is co-head of IRISA and Inria Rennes gender equality committee, and member of Inria's gender equality and equal opportunities committee.
- Loïc Hélouët is the president of the *Commission Personnel* (Temporary staff committee) at Inria Rennes, responsible for evaluation of hirings on non-permanent positions.
- Loïc Hélouët is elected member of the "Formation Specialisée (Health and security committee)" of Inria and secretary of this committee.
- Thierry Jéron is référent chercheur for Inria Rennes and IRISA.

11.2 Teaching Supervision Juries Educational and pedagogical outreach

11.2.1 Teaching

Almost all members of DEVINE do teach.

- Licence: Nathalie Bertrand, Algorithms II, 18h, ENS Rennes;
- BUT: Loïc Germerie Guizouarn, network and cyber-security classes, 246h, IUT Saint-Malo;
- Licence: Aline Goeminne, Algorithms I, 19.5h, ENS Rennes;
- Master: Aline Goeminne, Algorithms, 9h, Agrégation, ENS Rennes;
- Master: Aline Goeminne, Computability and Complexity, 10.5h, Agrégation, ENS Rennes;
- Master: Aline Goeminne, Formal languages, 12h, Agrégation, ENS Rennes;
- Master: Aline Goeminne, Algorithms and data structures, 16h, Agrégation, ENS Rennes.
- Master: Loïc Hélouët , Algorithms and proofs, 16h, Agrégation, ENS Rennes;
- Licence: Loïc Hélouët, Algorithms and Java, 40h, INSA Rennes;
- Licence: Julie Parreaux, Algorithms 2, 21h, ENS Rennes;
- Licence: Julie Parreaux, Formals Tools for Computer scientists, 52h, ISTIC, Université de Rennes;
- Licence: Julie Parreaux, Algorithms and Complexity, 47h, ISTIC, Université de Rennes;
- Licence: Julie Parreaux, Programming 2, 24h, ISTIC, Université de Rennes;
- Licence: Julie Parreaux, Logic, 16.5h, ISTIC, Université de Rennes;
- Master: Julie Parreaux, Validation and Verification, 16.5h, ISTIC, Université de Rennes.

11.2.2 Supervision

PhD Students

- PhD in Progress
  - Mathieu Laurent, on Efficient verification of asynchronous distributed systems, started in October 2023, supervised by Martin Quinson (Magellan) and Thierry Jéron;
  - Luc Lapointe (ENS Paris-Saclay), AGPR ENS Paris-Saclay, on concurrent games with parameterized number of participants, started in September 2023, supervised by Nathalie Bertrand and Patricia Bouyer (LMF);
  - Luca Paparazzo, ENS Grant, on Quantitative games in timed systems, application to energy savings in urban transport, started in October 2024, supervised by Nathalie Bertrand and Loïc Hélouët;
  - Aymeric Come, on Approximation methods for the soundness of control laws derived by machine learning, started in December 2022, supervised by Eric Fabre and Loïc Hélouët;
  - Pranav Ghorpade (Univ. Sydney), on Verification of distributed algorithms within blockchains, started in October 2024, supervised by Nathalie Bertrand and Sasha Rubin (Univ. Sydney).
  - Victorien Desbois, on Heuristic search algorithms for vehicle rescheduling problem, started in December 2023, supervised by Ocan Sankur, François Schwarzentruber, Cédric Pelois (NewLogUp).
- Past PhDs
  - Antoine Thébault, CIFRE Grant, on Efficient learning techniques for traffic management in Transport Networks, started in September 2022, supervised by Kenza Saiah (Alstom Transport) and Loïc Hélouët. PhD not defended.

Master Students

- Maëlle Gautrin, M2 student at ENS Paris-Saclay, has been supervised for her M2 research internship on the Robustness in Weighted Timed Games by Nathalie Bertrand and Julie Parreaux.

Undergraduate Students

- Nowar Kazem, L3 student at ENS Rennes, has been supervised by Loïc Germerie-Guizouarn and Thierry Jéron on the Decidability of deadlock-freedom of RSC systems.

11.2.3 Juries

PhD committees

- Nathalie Bertrand was on the PhD defense jury of Alexandre Terefenko (UMONS, Belgium, march 2025), James Main (UMONS, Belgium, sept. 2025), Lucie Guillou (Université Paris Cité, sept. 2025). She was reviewer and took part to the jury of the PhD of Mathias Déhais (Université Caen Normandie, nov. 2025).
- Loïc Hélouët was reviewer in the PhD jury of Sarah Larroze Jardiné at LABRI in Dec. 2025.

Hiring committees

- In 2025, Nathalie Bertrand has been on the hiring committee of a Professor position at INSA Rennes, a Professor position at ENS Paris-Saclay, and on the admissibility jury of CRCN-ISFP competition at Inria centre at the University Grenoble Alpes.

11.3 Popularization

11.3.1 Participation in Live events

- Loïc Hélouët contributed to the organization of a seminar on Livestorm for PhD and Postdocs at Inria Rennes (1h each, 40-50 participants), dedicated to Work environment for PhD students.

11.3.2 Others science outreach relevant activities

- Loïc Hélouët contributed to the CHICHE! program (15 classes visited).

12 Scientific production

12.1 Major publications

[1] S. Akshay, L. Hélouët and R. Phawade. 'Combining Free choice and Time in Petri Nets'. In: *Journal of Logical and Algebraic Methods in Programming* (18th May 2020), pp. 1–36. DOI: 10.1016/j.jlamp.2018.11.006. URL: https://inria.hal.science/hal-01931728.

[2] C. Baier, N. Bertrand, C. Dubslaff, D. Gburek and O. Sankur. 'Stochastic Shortest Paths and Weight-Bounded Properties in Markov Decision Processes'. In: LICS '18 - 33rd Annual ACM/IEEE Symposium on Logic in Computer Science. Oxford, United Kingdom: ACM Press, 9th July 2018, pp. 86–94. DOI: 10.1145/3209108.3209184. URL: https://hal.science/hal-01883409.

[3] N. Bertrand, M. Dewaskar, B. Genest, H. Gimbert and A. Godbole. 'Controlling a population'. In: *Logical Methods in Computer Science* 15.3 (2019), pp. 1–30. DOI: 10.23638/LMCS-15(3:6)2019. URL: https://hal.science/hal-02350251.

[4] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, J. Ouaknine and J. Worrell. 'Model Checking Real-Time Systems'. In: *Handbook of model checking*. Springer-Verlag, 2nd Apr. 2018, pp. 1001–1046. DOI: 10.1007/978-3-319-10575-8_29. URL: https://hal.science/hal-01889280.

[5] L. Henry, T. Jéron and N. Markey. 'Active learning of timed automata with unobservable resets'. In: FORMATS 2020 - 18th International Conference on Formal Modeling and Analysis of Timed Systems. Vienna, Austria, 1st Sept. 2020, pp. 1–26. URL: https://inria.hal.science/hal-02896517.

[6] V. Roussanaly, O. Sankur and N. Markey. 'Abstraction Refinement Algorithms for Timed Automata'. In: CAV 2019 - 31st International Conference on Computer Aided Verification. Vol. 11561. LNCS. New York, United States: Springer, 12th July 2019, pp. 22–40. DOI: 10.1007/978-3-030-25540-4_2. URL: https://hal.science/hal-02265808.

12.2 Publications of the year

International journals

[7] A. Amrane, H. Bazille, U. Fahrenberg and K. Ziemiański. 'Closure and decision properties for higher-dimensional automata'. In: *Theoretical Computer Science* 1036 (2025), p. 115156. DOI: 10.1016/j.tcs.2025.115156. URL: https://hal.science/hal-05278036.

[8] A. Côme, E. Fabre and L. Hélouët. 'A Floyd-Warshall Approach to Value Computation in Markov Decision Processes (Extended Version)'. In: *International Journal on Software Tools for Technology Transfer*. International Journal on Software Tools for Technology Transfer, special issue QUEST+Formats 2024 (2026). URL: https://inria.hal.science/hal-05447868. In press (cit. on p. 16).

[9] B. Monmege, J. Parreaux and P.-A. Reynier. 'Decidability of One-Clock Weighted Timed Games with Arbitrary Weights'. In: *Logical Methods in Computer Science* 21.1 (28th Jan. 2025). DOI: 10.46298/lmcs-21(1:8)2025. URL: https://hal.science/hal-04920306 (cit. on p. 14).

[10] B. Monmege, J. Parreaux and P.-A. Reynier. 'Playing Stochastically in Weighted Timed Games to Emulate Memory'. In: *Logical Methods in Computer Science* 21.1 (26th Feb. 2025). DOI: 10.46298/lmcs-21(1:19)2025. URL: https://hal.science/hal-04973633 (cit. on p. 14).

[11] O. Sankur. 'Automatic Assume-Guarantee Reasoning for Safety and Liveness Using Passive Learning'. In: *Formal Methods in System Design* 66 (9th July 2025), pp. 498–528. DOI: 10.1007/s10703-025-00484-3. URL: https://hal.science/hal-05450239.

[12] O. Sankur. 'Timed Automata Verification and Synthesis Via Finite Automata Learning'. In: *Journal of Automated Reasoning* 69.2 (13th June 2025), p. 15. DOI: 10.1007/s10817-025-09730-z. URL: https://hal.science/hal-05450244.

International peer-reviewed conferences

[13] A. Amrane, H. Bazille, U. Fahrenberg, L. Hélouët and P. Schlehuber-Caissier. 'Petri Nets and Higher-Dimensional Automata'. In: *Lecture Notes in Computer Science*. Petri Nets 2025 - 46th International Conference on Application and Theory of Petri Nets and Concurrency. Vol. 15714. Aubervillers, France: Springer, 2025, pp. 18–40. DOI: 10.1007/978-3-031-94634-9_2. URL: https://hal.science/hal-05013777 (cit. on p. 15).

[14] A. Anand, L. Germerie Guizouarn, T. Jéron, S. Mukherjee, S. Pinisetty and O. Sankur. 'Prompt Runtime Enforcement'. In: ATVA 2025 - International Symposium on Automated Technology for Verification and Analysis. Bengalore, India: Springer, 2025, pp. 1–22. URL: https://inria.hal.science/hal-05229564 (cit. on pp. 14, 17).

[15] É. André, S. Jacobs, S. L. Karra and O. Sankur. 'Parameterized Verification of Timed Networks with Clock Invariants'. In: FSTTCS 2025 - 45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science. Vol. 45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2025). Goa, India: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 9th Dec. 2025. DOI: 10.4230/LIPIcs.FSTTCS.2025.8. URL: https://hal.science/hal-05450304.

[16] N. Bertrand, P. Ghorpade, S. Rubin, B. Scholz and P. Subotić. 'Reusable Formal Verification of DAG-based Consensus Protocols'. In: *Proceedings of the 17th Symposium on Nasa Formal Methods*. NFM 2025 - 17th NASA Formal Methods Symposium. Vol. Lecture Notes in Computer Science. 15682. Williamsburg (Virginia), United States: Springer, 2025. DOI: 10.1007/978-3-031-93706-4_9. URL: https://inria.hal.science/hal-05472550 (cit. on p. 15).

[17] N. Bertrand, L. Hélouët, E. Lefaucheux and L. Paparazzo. 'Reachability in multi-agent transfer systems'. In: VMCAI 2026 - 27th International Conference on Verification, Model Checking, and Abstract Interpretation. LNCS. Rennes, France, 2026. URL: https://inria.hal.science/hal-05447890 (cit. on p. 16).

[18] V. Bruyère, B. Garhewal, G. A. Pérez, G. Staquet and F. W. Vaandrager. 'Active Learning of Mealy Machines with Timers'. In: *Quantitative Evaluation of Systems and Formal Modeling and Analysis of Timed Systems. QEST+FORMATS 2025. Lecture Notes in Computer Science*. QEST+FORMATS 2025 - Quantitative Evaluation of Systems and Formal Modeling and Analysis of Timed Systems. Vol. 16143. Lecture Notes in Computer Science. Aarhus, Denmark: Springer Nature Switzerland, 2nd Oct. 2025, pp. 42–61. DOI: 10.1007/978-3-032-05792-1_3. URL: https://hal.science/hal-05376642 (cit. on p. 14).

[19] K. Chatterjee, L. Doyen, J.-F. Raskin and O. Sankur. 'The Value Problem for Multiple-Environment MDPs with Parity Objective'. In: ICALP 2025 - 52nd EATCS International Colloquium on Automata, Languages, and Programming. Vol. 52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025). Aarhus, Denmark: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. DOI: 10.4230/LIPIcs.ICALP.2025.150. URL: https://hal.science/hal-05450279.

[20] L. Dartois, P. Gastin, L. G. Guizouarn and S. Krishna. 'Reversible Pebble Transducers'. In: *Leibniz International Proceedings in Informatics (LIPIcs)*. CONCUR 2025 - 36th International Conference on Concurrency Theory. Vol. 36th International Conference on Concurrency Theory (CONCUR 2025). 348. Aarhus, Denmark: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 18th Aug. 2025, 14:1–14:22. DOI: 10.4230/LIPIcs.CONCUR.2025.14. URL: https://hal.science/hal-05303982 (cit. on p. 15).

[21] M. Laurent, T. Jéron and M. Quinson. 'Towards Efficient Verification of Parallel Applications with Mc SimGrid'. In: *DisCoTec 2025 - 20th International Federated Conference on Distributed Computing Techniques*. FORTE 2025 - 45th International Conference on Formal Techniques for Distributed Objects, Components, and Systems. Lecture Notes in Computer Science. Lille, France: Springer, 2025, pp. 1–21. URL: https://hal.science/hal-05042973 (cit. on p. 15).

[22] A. Majumdar, S. Mukherjee and J.-F. Raskin. 'Learning Event-Recording Automata Passively'. In: ATVA 2025 - Automated Technology for Verification and Analysis. Vol. 16145. Lecture Notes in Computer Science. Bengaluru, India: Springer Nature Switzerland, 26th Oct. 2025, pp. 27–48. DOI: 10.1007/978-3-032-08707-2_2. URL: https://hal.science/hal-05372935 (cit. on p. 14).

[23] L. Paparazzo, L. Hélouët and N. Markey. 'Energy Transfer in Timed Cyclic Networks'. In: *Lecture Notes in Computer Science*. Petri Nets 2025 - 46th International Conference on Applications and Theory of Petri Nets and Concurrency. Vol. 15714. Lecture Notes in Computer Science. Paris, France: Springer Nature Switzerland, 8th June 2025, pp. 197–218. DOI: 10.1007/978-3-031-94634-9_10. URL: https://inria.hal.science/hal-05447415 (cit. on p. 16).

Conferences without proceedings

[24] L. Passemard, A. Amrane and U. Fahrenberg. 'Higher-Dimensional Automata: Extension to Infinite Tracks'. In: FSCD 2025 - 10th International Conference on Formal Structures for Computation and Deduction. Birmingham, United Kingdom: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. DOI: 10.4230/LIPIcs.FSCD.2025.31. URL: https://hal.science/hal-05274025.

Scientific book chapters

[25] N. Bertrand, P. Bouyer, T. Brihaye, P. Fournier and P. Vandenhove. 'Decisiveness for Countable MDPs and Insights for NPLCSs and POMDPs'. In: *Principles of Formal Quantitative Analysis*. Vol. 15760. Lecture Notes in Computer Science. Springer Nature Switzerland, 30th Aug. 2026, pp. 70–98. DOI: 10.1007/978-3-031-97439-7_3. URL: https://hal.science/hal-05344326 (cit. on p. 17).

Edition (books, proceedings, special issue of a journal)

[26] *Principles of Formal Quantitative Analysis: Essays Dedicated to Christel Baier on the Occasion of Her 60th Birthday*. Principles of Formal Quantitative Analysis. Vol. 15760. Lecture Notes in Computer Science. Aarhus (Danemark), Denmark: Springer Nature Switzerland, 2025. DOI: 10.1007/978-3-031-97439-7. URL: https://inria.hal.science/hal-05472556.

12.3 Cited publications

[27] R. Alur and D. L. Dill. 'A Theory of Timed Automata'. In: *Theoretical Computer Science* 126.2 (1994), pp. 183–235. DOI: 10.1016/0304-3975(94)90010-8 (cit. on p. 7).

[28] K. Apt and D. C. Kozen. 'Limits for automatic verification of finite-state concurrent systems'. In: *Information Processing Letters* 22.6 (May 1986), pp. 307–309. DOI: 10.1016/0020-0190(86)90071-2 (cit. on p. 9).

[29] S. Aronis, B. Jonsson, M. Lång and K. Sagonas. 'Optimal Dynamic Partial Order Reduction with Observers'. In: *TACAS'18*. Vol. 10806. LNCS. Springer, 2018, pp. 229–248. DOI: 10.1007/978-3-319-89963-3_14 (cit. on p. 9).

[30] E. Bartocci, Y. Falcone, A. Francalanza and G. Reger. 'Introduction to Runtime Verification'. In: *Lectures on Runtime Verification: Introductory and Advanced Topics*. Springer, 2018, pp. 1–33. DOI: 10.1007/978-3-319-75632-5_1 (cit. on p. 8).

[31] G. Behrmann, A. David, K. G. Larsen, J. Håkansson, P. Pettersson, W. Yi and M. Hendriks. 'UPPAAL 4.0'. In: *QEST'06*. IEEE Comp. Soc. Press, Sept. 2006, pp. 125–126. DOI: 10.1109/QEST.2006.59 (cit. on p. 7).

[32] N. Bertrand, P. Fournier and A. Sangnier. 'Playing with Probabilities in Reconfigurable Broadcast Networks'. In: *FoSSaCS'14*. Vol. 8412. LNCS. Springer, 2014, pp. 134–148. DOI: 10.1007/978-3-642-54830-7_9 (cit. on p. 9).

[33] N. Bertrand, I. Konnov, M. Lazic and J. Widder. 'Verification of Randomized Consensus Algorithms Under Round-Rigid Adversaries'. In: *CONCUR'19*. Vol. 140. LIPIcs. Leibniz-Zentrum für Informatik, 2019, 33:1–33:15. DOI: 10.4230/LIPIcs.CONCUR.2019.33 (cit. on p. 9).

[34] N. Bertrand, M. Lazic and J. Widder. 'A Reduction Theorem for Randomized Distributed Algorithms Under Weak Adversaries'. In: *VMCAI'21*. Vol. 12597. LNCS. Springer, 2021, pp. 219–239. DOI: 10.1007/978-3-030-67067-2_11 (cit. on p. 9).

[35] N. Bertrand, B. Thomas and J. Widder. 'Guard Automata for the Verification of Safety and Liveness of Distributed Algorithms'. In: *CONCUR'21*. Vol. 203. LIPIcs. Leibniz-Zentrum für Informatik, 2021, 15:1–15:17. DOI: 10.4230/LIPIcs.CONCUR.2021.15 (cit. on p. 9).

[36] B. Bonakdarpour, P. Prabhakar and C. Sánchez. 'Model Checking Timed Hyperproperties in Discrete-Time Systems'. In: *NFM'20*. Vol. 12229. LNCS. Springer, 2020, pp. 311–328. DOI: 10.1007/978-3-030-55754-6_18 (cit. on p. 8).

[37] P. Bouyer, L. Henry, S. Jaziri, T. Jéron and N. Markey. 'Diagnosing timed automata using timed markings'. In: *International Journal on Software Tools for Technology Transfer* 23.2 (Apr. 2021), pp. 229–253. DOI: 10.1007/s10009-021-00606-2 (cit. on p. 8).

[38] P. Bouyer, O. Kupferman, N. Markey, B. Maubert, A. Murano and G. Perelli. 'Reasoning about Quality and Fuzziness of Strategic Behaviours'. In: *ACM Transactions on Computational Logic* (2023). To appear (cit. on p. 10).

[39] R. Brenguier, G. A. Pérez, J. Raskin and O. Sankur. 'Admissibility in Quantitative Graph Games'. In: *FSTTCS'16*. Vol. 65. LIPIcs. Leibniz-Zentrum für Informatik, 2016, 42:1–42:14. DOI: 10.4230/LIPIcs.FSTTCS.2016.42 (cit. on p. 10).

[40] K. Chatterjee, T. A. Henzinger and N. Piterman. 'Strategy Logic'. In: *Information and Computation* 208.6 (June 2010), pp. 677–693. DOI: 10.1016/j.ic.2009.07.004 (cit. on p. 10).

[41] E. Clarke, D. Long and K. McMillan. 'Compositional model checking'. In: *LICS'89*. IEEE Comp. Soc. Press, 1989, pp. 353–362 (cit. on p. 7).

[42] M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe and C. Sánchez. 'Temporal Logics for Hyperproperties'. In: *POST'14*. Vol. 8414. LNCS. Springer, 2014, pp. 265–284. DOI: 10.1007/978-3-642-54792-8_15 (cit. on p. 8).

[43] M. R. Clarkson and F. B. Schneider. 'Hyperproperties'. In: *Journal of Computer Security* 18.6 (2010), pp. 1157–1210. DOI: 10.3233/JCS-2009-0393 (cit. on p. 8).

[44] E. Clement, T. Jéron, N. Markey and D. Mentré. 'Computing maximally-permissive strategies in acyclic timed automata'. In: *FORMATS'20*. Vol. 12288. LNCS. Springer, Sept. 2020, pp. 111–126. DOI: 10.1007/978-3-030-57628-8_7 (cit. on p. 8).

[45] K. E. Coons, M. Musuvathi and K. S. McKinley. 'Bounded partial-order reduction'. In: *OOPSLA'13*. ACM Press, 2013, pp. 833–848. DOI: 10.1145/2509136.2509556 (cit. on p. 9).

[46] A. D'Ariano, M. Pranzo and I. A. Hansen. 'Conflict Resolution and Train Speed Coordination for Solving Real-Time Timetable Perturbations'. In: *IEEE Transactions on Intelligent Transportation Systems* 8.2 (2007), pp. 208–222 (cit. on p. 10).

[47] A. D'Ariano, D. Pacciarelli and M. Pranzo. 'A branch-and-bound algorithm for scheduling trains in a railway network'. In: *European Journal of Operational Research* 183.2 (2007), pp. 643–657 (cit. on p. 10).

[48] L. Dartois, P. Fournier, I. Jecker and N. Lhote. 'On Reversible Transducers'. In: *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, Warsaw, Poland, July 10-14, 2017*. Ed. by I. Chatzigiannakis, P. Indyk, F. Kuhn and A. Muscholl. Vol. 80. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 113:1–113:12. DOI: 10.4230/LIPICS.ICALP.2017.113. URL: https://doi.org/10.4230/LIPIcs.ICALP.2017.113 (cit. on p. 15).

[49] C. Dehnert, D. Gebler, M. Volpato and D. N. Jansen. 'On Abstraction of Probabilistic Systems'. In: *ROCKS'12*. Vol. 8453. LNCS. Springer, 2012, pp. 87–116. DOI: 10.1007/978-3-662-45489-3_4 (cit. on p. 9).

[50] S. Edelkamp, V. Schuppan, D. Bosnacki, A. Wijs, A. Fehnker and H. Aljazzar. 'Survey on Directed Model Checking'. In: *MoChArt'08*. Vol. 5348. LNCS. Springer, 2008, pp. 65–89. DOI: 10.1007/978-3-642-00431-5_5 (cit. on p. 9).

[51] A. B. Eriksen, C. Huang, J. Kildebogaard, H. Lahrmann, K. G. Larsen, M. Muñiz and J. H. Taankvist. 'Uppaal Stratego for Intelligent Traffic Lights'. In: *12th ITS European Congress*. 2017 (cit. on p. 10).

[52] C. Flanagan and P. Godefroid. 'Dynamic partial-order reduction for model checking software'. In: *POPL'05*. ACM Press, 2005, pp. 110–121. DOI: 10.1145/1040305.1040315 (cit. on p. 9).

[53] S. Graf and H. Saidi. 'Construction of abstract state graphs with PVS'. In: *CAV'97*. Vol. 1254. LNCS. Springer. 1997, pp. 72–83 (cit. on p. 7).

[54] K. Havelund, A. Skou, K. G. Larsen and K. Lund. 'Formal modeling and analysis of an audio/video protocol: An industrial case study using UPPAAL'. In: *Proceedings Real-Time Systems Symposium*. IEEE. 1997, pp. 2–13 (cit. on p. 7).

[55] L. Hélouët, N. Markey and R. Raha. 'Reachability games with relaxed energy constraints'. In: *Information and Computation* 285 (Part B) (May 2022). DOI: 10.1016/j.ic.2021.104806 (cit. on p. 10).

[56] L. Henry, T. Jéron and N. Markey. 'Active Learning of Timed Automata with Unobservable Resets'. In: *FORMATS'20*. Vol. 12288. LNCS. Springer, Sept. 2020, pp. 144–160. DOI: 10.1007/978-3-030-57628-8_9 (cit. on p. 8).

[57] L. Henry, T. Jéron and N. Markey. 'Control strategies for off-line testing of timed systems'. In: *Formal Methods in System Design* (2023). DOI: 10.1007/s10703-022-00403-w (cit. on p. 8).

[58] F. Herbreteau and G. Point. *TChecker*. https://github.com/ticktac-project/tchecker. 2019 (cit. on pp. 7, 8).

[59] H. Hermanns, B. Wachter and L. Zhang. 'Probabilistic CEGAR'. In: *CAV'08*. Vol. 5123. LNCS. Springer, 2008, pp. 162–175. DOI: 10.1007/978-3-540-70545-1_16 (cit. on p. 9).

[60] A. Horváth, M. Paolieri, L. Ridi and E. Vicario. 'Transient analysis of non-Markovian models using stochastic state classes'. In: *Performance Evaluation* 69.7-8 (2012), pp. 315–335 (cit. on p. 10).

[61] T. Jéron, N. Markey, D. Mentré, R. Noguchi and O. Sankur. 'Incremental methods for checking real-time consistency'. In: *FORMATS'20*. Vol. 12288. LNCS. Springer, Sept. 2020, pp. 249–264. DOI: 10.1007/978-3-030-57628-8_15 (cit. on p. 8).

[62] K. Lampka, S. Perathoner and L. Thiele. 'Analytic real-time analysis and timed automata: a hybrid method for analyzing embedded real-time systems'. In: *Proceedings of the seventh ACM international conference on Embedded software*. 2009, pp. 107–116 (cit. on p. 7).

[63] D. J. Lehmann and M. O. Rabin. 'On the Advantages of Free Choice: A Symmetric and Fully Distributed Solution to the Dining Philosophers Problem'. In: *POPL'81*. ACM Press, 1981, pp. 133–138. DOI: 10.1145/567532.567547 (cit. on p. 6).

[64] Z. Luo, M. Zheng and S. F. Siegel. 'Verification of MPI programs using CIVL'. In: *EuroMPI'17*. ACM Press, 2017, 6:1–6:11. DOI: 10.1145/3127024.3127032 (cit. on p. 9).

[65] F. Martinelli, F. Mercaldo, A. Santone, C. Tavolato-Wötzl and P. Tavolato. *Timed Automata Networks for SCADA Attacks Real-Time Mitigation*. 2019 (cit. on p. 7).

[66] T. A. Pham, T. Jéron and M. Quinson. 'Unfolding-Based Dynamic Partial Order Reduction of Asynchronous Distributed Programs'. In: *FORTE 2019*. Vol. 11535. LNCS. Springer, 2019, pp. 224–241. DOI: 10.1007/978-3-030-21759-4_13 (cit. on p. 9).

[67] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014 (cit. on p. 10).

[68] C. Rodríguez, M. Sousa, S. Sharma and D. Kroening. 'Unfolding-based Partial Order Reduction'. In: *CONCUR'15*. Vol. 42. LIPIcs. Leibniz-Zentrum für Informatik, 2015, pp. 456–469. DOI: 10.4230/LIPIcs.CONCUR.2015.456 (cit. on p. 9).

[69] V. Roussanaly, O. Sankur and N. Markey. 'Abstraction Refinement Algorithms for Timed Automata'. In: *CAV'19*. Vol. 11561. LNCS. Springer, July 2019, pp. 22–40. DOI: 10.1007/978-3-030-25540-4_2 (cit. on p. 7).

[70] O. Sankur and B. Thomas. 'PyLTA: A Verification Tool for Parameterized Distributed Algorithms'. In: *TACAS'23*. Vol. 13994. LNCS. Springer, 2023 (cit. on p. 9).

[71] H. A. Simon. 'Rational Choice and the Structure of the Environment'. In: *Psychological Review* 63.2 (1956), pp. 129–138 (cit. on p. 10).

[72] A. Tamatsukuri and T. Takahashi. 'Guaranteed satisficing and finite regret: Analysis of a cognitive satisficing value function'. In: *Biosystems* 180 (2019), pp. 46–53 (cit. on p. 10).

[73] M. Utting, A. Pretschner and B. Legeard. 'A taxonomy of model-based testing approaches'. In: *Software Testing, Verification and Reliability* 22.5 (2012), pp. 297–312. DOI: 10.1002/stvr.456 (cit. on p. 8).