RESEARCH CENTRE

Inria Centre at Rennes University

IN PARTNERSHIP WITH:

CNRS, Université de Rennes

2024

ACTIVITY REPORT

Project-Team

DEVINE

DEpendable distributed systems: formal VerificatIoN made Efficient

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Proofs and Verification

Inria

Contents

Project-Team DEVINE

Creation of the Project-Team: 2024 January 01

Keywords

Computer sciences and digital sciences

A2.4. – Formal method for verification, reliability, certification

A2.4.2. – Model-checking

A2.5.5. – Software testing

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

1 Team members, visitors, external collaborators

Research Scientists

- Nathalie Bertrand [Team leader, INRIA, Senior Researcher]
- Loïc Hélouët [INRIA, Senior Researcher]
- Thierry Jéron [INRIA, Senior Researcher]
- Nicolas Markey [CNRS, Senior Researcher, until Aug 2024]
- Ocan Sankur [CNRS, Researcher, until Sep 2024]

Faculty Members

- Loïc Germerie-Guizouarn [UNIV RENNES, Associate Professor, from Sep 2024]
- Julie Parreaux [UNIV RENNES, Associate Professor, from Sep 2024]

Post-Doctoral Fellow

- Gaëtan Staquet [INRIA, Post-Doctoral Fellow, from Oct 2024]

PhD Students

- Aymeric Côme [INRIA]
- Victorien Desbois [NewLogUP]
- Pranav Ghorpade [Univ. Sydney, from Oct 2024, located in Sydney]
- Luc Lapointe [ENS Paris-Saclay, located at LMF]
- Mathieu Laurent [ENS RENNES]
- Luca Paparazzo [ENS RENNES, from Oct 2024]
- Antoine Thébault [ALSTOM, CIFRE]
- Nicolas Waldburger [UNIV RENNES]

Interns and Apprentices

- Orane Belhomme [ENS DE LYON, Intern, from Jun 2024 until Jul 2024]
- Luca Paparazzo [INRIA, Intern, from Feb 2024 until Jul 2024]

Administrative Assistant

- Laurence Dinh [INRIA]

External Collaborator

- Ulrich Fahrenberg [EPITA]

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 [56]: “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 distributed systems with quantitative aspects**. To achieve this objective, we will

develop model-based[1] scalable verification and optimization techniques.

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 [21]. 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 [25, 51]. 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 [47, 55, 58] 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.

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

In this context, predicate abstraction [46, 62] 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 [35]; 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 [51].

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. [66]): 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 [50] and consistency checking [54]; 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 [24] 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 [31], 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* [38] 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 [49, 50], 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 [37, 36], 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 [30]. 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 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 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 [59] 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) [45, 23, 61], 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 [43]. Directed model checking prioritizes the state-space exploration using A*-like 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 [39] 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 [22], 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 [57].

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 [26, 27, 28]. 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 [29, 63]. Building on the latter, and on an existing CEGAR framework for finite-state probabilistic systems [52, 42], 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 [60] 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 [34]. 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 [32] 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 [64] 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. [65]). 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 [33, 48], 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 [41], 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 [40]. 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 [44]. Symbolic representations of sets of states and distributions [53] 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

The DEVINE team participated in the Extended Stay Support Scheme (ESSS) set up by TCS4F (Theoretical Computer Science for Future) during the international event Highlights that took place in Bordeaux in summer 2024. The ESSS aimed at exploiting the presence of distant researchers (from Asia, America, etc.) at these conferences to invite them over for a longer stay in France or Europe. More generally, some members of the team individually take part in the TCS4F 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 carefully choose 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

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.

6 Highlights of the year

- DEVINE was created on January 1st 2024!
- DEVINE is happy to welcome two new permament members: Julie Parreaux and Loïc Germerie-Guizouarn, both assistant professor at Univ. Rennes.
- Nicolas Markey left the team in August 2024 to become a middle-school maths teacher.
- The team has organized the summer school MOVEP in May 2024, gathering approximately 40 people (PhD and post-docs mostly) working in formal verification.

7 New software, platforms, open data

7.1 New software

7.1.1 MOCHY

Name: MOdels for Concurrent and HYbrid systems

Keywords: Public transport, Hybrid models, Simulation, Performance analysis

Functional Description: Allows for the modeling of hybrid systems, schedule and regulation algorithms to optimize Key Performance Indicators. Mochy addresses mainly models of transport networks, their timetables and traffic management techniques. The tool allows for the fast simulation of these regulated models, to measure performance indicators. Since version 2.0, MOCHY proposes a novel traffic management algorithm with neural networks.

Release Contributions: Co-simulation of time Petri nets and timetables (model for regulated urban train systems with a hold-on policy). Performance analysis for simple Key Performance Indicators. Traffic management with neural networks.

URL: https://adt-mochy.gitlabpages.inria.fr/mochy/

Contact: Loic Helouet

7.1.2 PyLTA

Keywords: Model Checking, Distributed computing

Functional Description: PyLTA is written in Python. It uses an ad hoc input format to represent distributed algorithms and their specification. The verification process builds on the counter-example guided abstraction refinement (CEGAR) principle.

URL: https://gitlab.com/BastienT/pylta

Publication: hal-03996060

Contact: Nathalie Bertrand

Participants: Bastien Thomas, Ocan Sankur, Nathalie Bertrand

8 New results

8.1 New results on Efficient analysis of real-time systems

Participants: Aymeric Côme, Loïc Hélouët, Thierry Jéron, Nicolas Markey, Ocan Sankur.

Timed extensions of Petri nets In [13] we have considered verification of timed models handling additional quantities progressing linearly such as distance of moving objects to a target. We introduced a variant of Petri nets called trajectory nets where some places are standard control places containing tokens, and other places contain a trajectory of an object. We give a semantics for this model, and propose an abstraction of sets of equivalent trajectories into symbolic domains. These domains cannot be represented by Difference Bound Matrices, but one can compute in polynomial time a symbolic representation of successor configurations. Furthermore domains are closed under this successor relation, and the set of domains of a trajectory net is finite. A consequence is that, when the control part of a trajectory net is bounded, reachability, coverability and verification of safety properties involving distances are PSPACE-Complete.

In [7], we have extended results on a model called Waiting nets. In time Petri nets (TPNs), time and control are tightly connected: time measurement for a transition starts only when all resources needed to fire it are available. Further, upper bounds on duration of enabledness can force transitions to fire (this is called urgency). For many systems, one wants to decouple control and time, i.e. start measuring time as soon as a part of the preset of a transition is filled, and fire it after some delay and when all needed resources are available. Waiting nets are an extension of TPN that dissociates time measurement and control. Their semantics allows time measurement to start with incomplete presets, and can ignore urgency when upper bounds of intervals are reached but all resources needed to fire are not yet available. Firing of a transition is then allowed as soon as missing resources are available. It is known that extending bounded TPNs with stopwatches leads to undecidability. Our extension is weaker, and we show in this work how to compute a finite state class graph for bounded waiting nets, yielding decidability of reachability and coverability. Last, we have compared expressiveness of waiting nets with that of other models w.r.t. timed language equivalence, and show that waiting nets are strictly more expressive than TPNs.

Efficient evaluation of policies in Markov decision processes In [10] we have revisited the value iteration question for Markov Decision 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 work 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. The same formalism also gives access to other quantities of interest, such as the gradient of the average reward with respect to model or policy parameters, or the variance of the reward. The behaviors and performance of this value estimation scheme are illustrated on several benchmarks.

Runtime monitoring of timed properties In formal verification, runtime monitoring consists in observing the execution of a system in order to decide as quickly as possible whether or not it satisfies a given property. We consider monitoring in a distributed setting, for properties given as reachability timed automata. In such a setting, the system is made of several components, each equipped with its own local

clock and monitor. The monitors observe events occurring on their associated component, and receive timestamped events from other monitors through FIFO channels. Since clocks are local, they cannot be perfectly synchronized, resulting in imprecise timestamps. Consequently, they must be seen as intervals, leading monitors to consider possible reorderings of events. In this context, each monitor aims to provide, as early as possible, a verdict on the property it is monitoring, based on its potentially incomplete and imprecise knowledge of the current execution. In [14], we propose an online monitoring algorithm for timed properties, robust to time imprecision and partial information from distant components. We first identify the date at which a monitor can safely compute a verdict based on received events. We then propose a monitoring algorithm that updates this date when new information arrives, maintains the current set of states in which the property can reside, and updates its verdict accordingly.

Test synthesis from requirements using Monte-Carlo tree search In [19] we consider the automatic online synthesis of black-box test cases from functional requirements specified as automata for reactive implementations. The goal of the tester is to reach some given state, so as to satisfy a coverage criterion, while monitoring the violation of the requirements. We develop an approach based on Monte Carlo Tree Search, which is a classical technique in reinforcement learning for efficiently selecting promising inputs. Seeing the automata requirements as a game between the implementation and the tester, we develop a heuristic by biasing the search towards inputs that are promising in this game. We experimentally show that our heuristic accelerates the convergence of the Monte Carlo Tree Search algorithm, thus improving the performance of testing.

8.2 Verification of distributed algorithms

Participants: Nathalie Bertrand, Pranav Ghorpade, Thierry Jéron, Nicolas Waldburger.

Model-checking models for distributed algorithms This year, we pursued our effort in establishing the decidability and complexity of model-checking problems for models of distributed systems. On the one hand, we studied the verification of population protocols with unordered data; on the other hand, we considered the extension of the well-known model of broadcast networks with registers.

Population protocols are a well-studied model of distributed computation in which a group of anonymous finite-state agents communicates via pairwise interactions. Together they decide whether their initial configuration, i. e., the initial distribution of agents in the states, satisfies a property. As an extension in order to express properties of multisets over an infinite data domain, Blondin and Ladouceur (ICALP'23) introduced population protocols with unordered data (PPUD). In PPUD, each agent carries a fixed data value, and the interactions between agents depend on whether their data are equal or not. Blondin and Ladouceur also identified the interesting subclass of immediate observation PPUD (IOPPUD), where in every transition one of the two agents remains passive and does not move, and they characterised its expressive power. We study the decidability and complexity of formally verifying these protocols. The main verification problem for population protocols is well-specification, that is, checking whether the given PPUD computes some function. We show that well-specification is undecidable in general. By contrast, for IOPPUD, we exhibit a large yet natural class of problems, which includes well-specification among other classic problems, and establish that these problems are in ExpSpace. We also provide a lower complexity bound, namely coNExpTime-hardness. [8]

In [12], we consider the parameterized verification of arbitrarily large networks of agents which communicate by broadcasting and receiving messages. In our model, the broadcast topology is reconfigurable so that a sent message can be received by any set of agents. In addition, agents have local registers which are initially distinct and may therefore be thought of as identifiers. When an agent broadcasts a message, it appends to the message the value stored in one of its registers. Upon reception, an agent can store the received value or test this value for equality with one of its own registers. We consider the coverability problem, where one asks whether a given state of the system may be reached by at least one agent. We establish that this problem is decidable; however, it is as hard as coverability in lossy channel systems, which is non-primitive recursive. This model lies at the frontier of decidability as other classical problems

on this model are undecidable; this is in particular true for the target problem where all processes must synchronize on a given state. By contrast, we show that the coverability problem is NP-complete when each agent has only one register.

Foundations of parameterized verification Since parameterized verification questions are often related to counter systems, we studied invariants for one-counter automata with disequality tests in [9]. A disequality test is a guard that prohibits a specified counter value. This reachability problem has been known to be NP-hard and in PSPACE, and characterising its computational complexity has been left as a challenging open question by Almagor, Cohen, Pérez, Shirmohammadi, and Worrell (2020). We reduce the complexity gap, placing the problem into the second level of the polynomial hierarchy, namely into the class $coNP^{NP}$. In the presence of both equality and disequality tests, our upper bound is at the third level, $P^{NP^{NP}}$. To prove this result, we show that non-reachability can be witnessed by a pair of invariants (forward and backward). These invariants are almost inductive. They aim to over-approximate only a "core" of the reachability set instead of the entire set. The invariants are also leaky: it is possible to escape the set. We complement this with separate checks as the leaks can only occur in a controlled way.

Formal verification of blockchains On a more practical side, we aim at designing verification techniques for actual code or low-level models of distributed algorithms. Towards this goal, we propose a generic approach to the formal verification of consensus protocols within blockchains. DAG-based consensus protocols are being adopted by blockchain companies to decrease energy footprints and improve security. A DAG-based consensus protocol collaboratively constructs a partial order of blocks of transactions and produces linearly ordered blocks. The ubiquity and strategic importance of blockchains call for formal proofs of correctness of key components, namely, consensus protocols. The preprint Reusable Formal Verification of DAG-based Consensus Protocols presents a safety-proven formal specification of two DAG-based protocols. Our specification highlights several dissemination, DAG construction, and ordering variations that can be combined to express the two protocols. The formalization requires a refinement approach for modeling the consensus. In an abstract model, we first show the safety of DAG-based consensus on leader blocks and then further refine the specification to encompass all blocks for all processes. The TLA+ specification for a given protocol consists of 492-732 lines, and the proof system TLAPS verifies 2025-2294 obligations in 6-8 minutes.

8.3 Optimization of multi-agent systems

Participants: Nathalie Bertrand, Loïc Hélouët, Nicolas Markey, Ocan Sankur, Antoine Thébault.

Quantitative evaluation and optimization of metro regulation policies Members of DEVINE have a long-standing bilateral collaboration with Alstom on urban train systems regulation.

In [15] we consider simulation models for Metro systems to measure useful information that guides optimization of traffic management strategies w.r.t. goals such as satisfaction of passenger demands, adherence to schedules or energy saving. Many network models are too precise for the analysis needs, and do not exploit concurrency. This results in an explosion in the size of models, and long simulation times. The model proposed in this work is an extension of Petri nets that handles trajectories of trains, passenger flows, and scenarios for passenger arrivals. We then define a fast event-based simulation scheme. We have tested our model on a real case study, the Metro of Montreal, and shown that full days of train operations with passengers can be simulated in a few seconds, allowing analysis of quantitative properties.

In [20] we have proposed algorithms for online adaptation of Metro timetables. Metro networks are usually operated with timetables, i.e. schedules with fixed dates for departure and arrival events. However, when a delay occurs, timetables have to be adapted to mitigate the effect of this time gap. We have proposed several algorithms to propagate the effects of primary delays in a timetable. The principle of delay propagation is to compute the minimal modification to the original timetable induced

by this delay. We define timetables as weighted acyclic graphs depicting events, causal dependencies, and timing constraints, and show that the minimal modification is achieved by rescheduling events as soon as possible in this graph. We have proposed five algorithms to compute efficiently new consistent dates after a delay, with the minimal modification w.r.t. the original timetable. The first algorithm uses properties of critical paths in the timetable, the second algorithm builds a topological ordering on events before a linear rescheduling of events dates. The third algorithm is a recursive scheme that may explore the whole timetable in the worst case, but stops on events that need no rescheduling. The next algorithms are still recursive schemes, but use heuristics to choose an ordering for events updates. Recursive schemes have a bad theoretical worst case complexity. However, tests on two real case studies show that they are efficient in practice, and reschedule timetables in a fraction of a second. This work has been submitted to a journal dedicated to transportation.

Planification in multi-agent systems In [11] we present a new algorithm for solving the connected multi-agent path finding problem (connected MAPF) which consists in finding paths for a set of agents that avoid collisions but also ensure connectivity between agents during the mission. Our algorithm is based on heuristic search and combines ODrM*, a well-known algorithm without connectivity constraints, and an efficient but incomplete solver for the connected MAPF from the literature. We present a formal analysis of the termination and completeness of our algorithm, and present an experimental evaluation, showing a significant improvement over the state of the art.

Foundations of games Games are a well-studied mathematical model to represent the interactions of agents in multi-agent systems. The games considered in the verification community often are games played on graphs. They form a prominent model in two related fields: the first being automata theory and logic, and the second verification and synthesis, both of which have been very active for decades. Members of the DEVINE team contributed to a textbook [16] presenting the state of the art on games on graphs from automata and logic. Specifically, we wrote the chapters on stochastic games, on timed games and on multi-player games.

Logics and automata In the preprint Arbitrary-arity Tree Automata and QCTL we introduce a new class of automata (which we coin EU-automata) running on infinite trees of arbitrary (finite) arity. We develop and study several algorithms to perform classical operations (union, intersection, complement, projection, alternation removal) for those automata, and precisely characterise their complexities. We also develop algorithms for solving membership and emptiness for the languages of trees accepted by EU-automata. We then use EU-automata to obtain several algorithmic and expressiveness results for the temporal logic QCTL (which extends CTL with quantification over atomic propositions) and for MSO. On the one hand, we obtain decision procedures with optimal complexity for QCTL satisfiability and model checking; on the other hand, we obtain an algorithm for translating any QCTL formula with k quantifier alternations to formulas with at most one quantifier alternation, at the expense of a (k+1)-exponential blow-up in the size of the formulas. Using the same techniques, we prove that any MSO formula can be translated into a formula with at most four quantifier alternations (and only two second-order-quantifier alternations), again with a (k+1)-exponential blow-up in the size of the formula.

9 Bilateral contracts and grants with industry

9.1 Bilateral contracts with industry

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. In 2022, Alstom and DEVINE stated a CIFRE PhD on the topic of smart traffic

management (PhD of Antoine Thébault, ANRT grant 2022-0444). The two main objectives are to 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. One of the key concerns in this collaboration is energy saving.

Mitsubishi Electric Research Center Europe (MERCE).

Participants: Thierry Jéron, , Nicolas Markey, , Ocan Sankur.

Several researchers of DEVINE have been involved in a collaboration on the verification of real-time systems with the "Information and Network Systems" Team (INSv) led by David Mentré of the "Communication & Information Systems (CIS)" Division of MERCE Rennes. The members of the team at MERCE have worked on different aspects of formal analysis of timed requirements. This year we focused on test generation using games and reinforcement learning from untimed requirements. A paper and a patent have been submitted. The collaboration finished in spring 2024.

10 Partnerships and cooperations

10.1 International initiatives

10.1.1 Inria associate team not involved in an IIL or an international program

DST/INRIA Associate Team SINCRET: Scalable and INCREmental security monitoring and enforcement for Timed systems

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

Website: https://devine.inria.fr/SINCRET/

Partner Institution: IIT Bhubaneswar, India

Led by Thierry Jéron and Srinivas Pinisetty

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. We currently propose 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 common publication will soon be submitted.

10.1.2 Participation in other International Programs

We are participating in the activities of the CNRS UMI ReLaX Research Lab in Computer Science, an Indo-French joint research unit dedicated to research in theoretical computer science, its applications and its interactions with mathematics. Within this setting, we are frequently hosting undergraduate students as interns from Indian universities such as Chennai Mathematical Institute, and IIT Bombay.

10.2 International research visitors

Ocan Sankur has been collaborating with Swen Jacobs (CISPA, Germany) on parameterized verification, and with Krishnendu Chatterjee (IST Austria) and Jean-François Raskin (ULB, Belgium) on planning and control problems on Markov decision processes.

Julie Parreaux collaborates with Sławomir Lasota (University of Warsaw, Poland) on the synthesis of timed automata.

Loïc Germerie Guizouarn collaborates with S. Krishna (IIT, Bombay) and R. Govind (Uppsala University, Sweden) on transducer models and verification of transformations.

10.3 National initiatives

10.3.1 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 Nicolas Markey until August 2024, then 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 mathematics, 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 is 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.

10.3.2 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 (LACL, Université Paris-Est Créteil) and Paul Gastin (LMF, Université Paris Saclay) on transducer models and verification of transformations;
- Victor Roussalany (Université de Lorraine) and Léo Henry (University College London, UK) on distributed monitoring of timed properties.
- François Laroussinie (IRIF, Univ. Paris-Cité) on arbitrary-arity tree automata for temporal logics.

11 Dissemination

Participants: Nathalie Bertrand, , Loïc Germerie-Guizouarn, , Loïc Hélouët, , Thierry Jéron, , Nicolas Markey, , Julie Parreaux, , Ocan Sankur.

11.1 Promoting scientific activities

11.1.1 Scientific events: organisation

Nicolas Markey, Nathalie Bertrand –for the scientific part– together with Laurence Dinh, Aymeric Côme, Nicolas Waldburger and Mathieu Laurent for the organisation part, hosted the MOVEP 2024 summer school in May. It gathered 40 PhD and postdocs from Europe to follow courses on the topic of verification.

11.1.2 Scientific events: selection

Member of the conference program committees

- Nathalie Bertrand has served as PC member of the international conferences VMCAI'24 and FoSSaCS'25.
- Loïc Hélouët has served as PC member of the international conference Petri Nets 2024.
- Ocan Sankur was in the program committee of ATVA'24, QEST-FORMAT'24, and GandALF'24.
- Thierry Jéron was in the program commitee of ICTSS'24.

Reviewer All members of the team reviewed a number of papers for various international conferences.

11.1.3 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.

11.1.4 Invited talks

Nathalie Bertrand was a panelist in the roundtable during the Women in Computer Science Workshop Cameroon in december 2024.

11.1.5 Leadership within the scientific community

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

11.1.6 Research administration

- Nicolas Markey has been the head of Department "Language and Software Engineering" of IRISA until August 2024.
- Loïc Hélouët was member of the "commission Personnel" (Temporary staff committee) responsible for evaluation of PhD hirings for Inria Rennes until September 2024. Since September 2024, he is the president of the commitee.

11.2 Teaching - Supervision - Juries

11.2.1 Teaching

Almost all members of DEVINE do teach, even the ones who have no teaching duties.

- Master: Nathalie Bertrand, Algorithms, and Symbolic AI, 18h, Agrégation, ENS Rennes, France;
- Master: Loïc Hélouët, Algorithms and proofs, 16h, Agrégation, ENS Rennes;
- Master: Nicolas Markey, Algorithms, 18h, Agrégation, ENS Rennes, France;
- Master: Nicolas Markey, Computability and Complexity, 18h, Agrégation, ENS Rennes, France;
- Licence: Loïc Hélouët, Algorithms and Java, 40h, INSA Rennes, France;
- BUT: Loïc Germerie Guizouarn, Networks Principles and Architecture, 20h, IUT Saint-Malo, France;
- Licence: Julie Parreaux, Formals Tools for Computer scientists, 52h, ISTIC, Université de Rennes, France;
- Licence: Julie Parreaux, Algorithms and Complexity, 47h, ISTIC, Université de Rennes, France;

11.2.2 Supervision

PhD students
Completed PhD

- Nicolas Waldburger, Parameterized verification of distributed algorithms under fairness conditions, started in Oct. 2021, defended in Dec. 2024, supervised by Nathalie Bertrand, Nicolas Markey and Ocan Sankur.

PhD in progress

- Aymeric Côme, Approximation methods for the soundness of control laws derived by machine learning, supervised by Éric Fabre and Loïc Hélouët;
- Antoine Thébault, CIFRE grant, Efficient learning techniques for management of transport systems, supervised by Loïc Hélouët;
- Luc Lapointe, AGPR ENS Paris-Saclay, on concurrent games with parameterized number of participants, started in September 2023, supervised by Nathalie Betrand and Patricia Bouyer (LMF);
- Luca Paparazzo, ENS Rennes grant, started in Oct. 2024, supervised by Loïc Hélouët and Nathalie Bertrand;
- Victorien Desbois, on Heuristic search algorithms for vehicle rescheduling problem, started in December 2023, supervised by Ocan Sankur, François Schwarzentruber, Cédric Pelois (NewLogUp).
- Thibaut Le Marre, on Multi-Agent Planning and Reinforcement Learning in an Epistemic Setting, started in October 2023, supervised by François Schwarzentruber, Jilles Dibangoye, Ocan Sankur.
- Mathieu Laurent, on Efficient verification of asynchronous distributed systems, started in October 2023, supervised by Martin Quinson (Myriads/Magellan) and Thierry Jéron;
- Pranav Ghorpade (Univ. Sydney), on verification of distributed algorithms within blockchains, started in October 2024, supervised by Nathalie Bertrand and Sasha Rubin (Univ. Sydney).

Master Students

- Luca Paparazzo, M2 student at ENS Rennes, has been supervised by Loïc Hélouët. His internship topic addresses energy saving in Metro systems using new energy games models.
- Gaëtan Regaud, M1 student at ENS Rennes, has been supervised by Uli Fahrenberg and Nicolas Markey on history-determinism for timed systems.
- Hugo Francon, prelab student at ENS Rennes, has been supervised by Nathalie Bertrand, Thierry Jéron and Nicolas Waldburger, on parameterized verification of asynchronous distributed systems.

Undergraduate Students

- Orane Belhomme, L3 student at ENS Lyon, has been supervised by Nicolas Markey.

11.2.3 Juries

Habilitation and PhD committees

- Nathalie Bertrand was on the PhD defense committee of Corto Mascle (Université de Bordeaux).
- Loïc Hélouët was reviewer for the PhD Thesis of Loriane Leclerq (Ecole Centrale de Nantes), and president of the Jury.
- Ocan Sankur was in the PhD defense committee of Gaëtan Staquet (Université de Mons).

Hiring committees

- Nathalie Bertrand was member of MCF hiring committees for Université de Bordeaux, Université Paris Nord and Sorbonne Université.
- Ocan Sankur was a member of the hiring committee for maître des conférences positions in Université de Rennes.
- Loïc Hélouët was member of the CRCN/ISFP hiring committee of Inria Rennes in May 2024. He was also member of the hiring committee for PhD students and Engineers in the "Moyens Incitatifs" program of Inria Rennes. Nicolas Markey was a member of the hiring committees for a MCF position at IUT Nantes/LS2N and at IUT Vannes/IRISA.

11.3 Other responsabilities

- Thierry Jéron is référent chercheur for Inria Rennes and IRISA.
- Nicolas Markey was co-head of the gender-equality committee of Inria Rennes and IRISA until August 2024, and Nathalie Bertrand since September 2024.
- Nathalie Bertrand is member of the "Formation Spécialisée de Site" of Inria for Rennes research center.
- Nathalie Bertrand is member of the Inria Parity and Equity committee (comité Parité et Égalité des chances).
- Loïc Hélouët is elected member of the "Formation Specialisée (Health and security committee)" of Inria and suppletive secretary of this committee.

11.4 Popularization

- Loïc Hélouët contributed to the organization of two seminars on Livestorm for PhD and Postdocs at Inria Rennes (1h each, 40-50 participants). The first one was dedicated to software development in the lab, the second one to scientific mediation. He was also invited to a national Livestorm to promote the Chiche! initiative.
- Nicolas Markey took part in the Chiche! Inria program.

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] L. Hélouët and P. Agrawal. 'Waiting Nets: State Classes and Taxonomy'. In: *Fundamenta Informaticae* 190.2-4 (30th Jan. 2024), pp. 63–107. DOI: 10.3233/FI-242167. URL: https://inria.hal.science/hal-04711522 (cit. on p. 10).

International peer-reviewed conferences

[8] S. van Bergerem, R. Guttenberg, S. Kiefer, C. Mascle, N. Waldburger and C. Weil-Kennedy. 'Verification of Population Protocols with Unordered Data'. In: ICALP 2024 - 51st EATCS International Colloquium on Automata, Languages and Programming. Tallinn, Estonia: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, pp. 1–40. DOI: 10.4230/LIPICS.ICALP.2024.156. URL: https://hal.science/hal-04707329 (cit. on p. 11).

[9] D. Chistikov, J. Leroux, H. Sinclair-Banks and N. Waldburger. 'Invariants for One-Counter Automata with Disequality Tests'. In: *Proceedings of the 35th International Conference on Concurrency Theory*. CONCUR2024 - International Conference on Concurrency Theory. Vol. 311. Calgary, Canada: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, pp. 1–34. DOI: 10.4230/LIPICS.CONCUR.2024.17. URL: https://hal.science/hal-04707413 (cit. on p. 12).

[10] A. Côme, É. Fabre and L. Hélouët. 'A Floyd-Warshall Approach to Value Computation in Markov Decision Processes'. In: *LNCS*. QEST-Formats 2024 - International Conference on Quantitative Evaluation of SysTems. Vol. 14996. Lecture Notes in Computer Science. Calgary, Canada: Springer Nature Switzerland; Springer Nature Switzerland, 29th Aug. 2024, pp. 284–301. DOI: 10.1007/978-3-031-68416-6_17. URL: https://inria.hal.science/hal-04711531 (cit. on p. 10).

[11] V. Desbois, O. Sankur and F. Schwarzentruber. 'An Efficient Modular Algorithm for Connected Multi-Agent Path Finding'. In: ECAI 2024 - 27th European Conference on Artificial Intelligence. Santiago de Compostela, Spain, 2024, pp. 1–11. URL: https://hal.science/hal-04650916 (cit. on p. 13).

[12] L. Guillou, C. Mascle and N. Waldburger. 'Parameterized Broadcast Networks with Registers: from NP to the Frontiers of Decidability'. In: *Foundations of Software Science and Computation Structures*. FoSSaCS 2024 - 27th International Conference on Foundations of Software Science and Computation Structures. Vol. 14575. Lecture Notes in Computer Science. Luxembourg, Luxembourg: Springer Nature Switzerland, 2024, pp. 250–270. DOI: 10.1007/978-3-031-57231-9. URL: https://hal.science/hal-04569794 (cit. on p. 11).

[13] L. Hélouët and P. Contractor. 'Symbolic domains and reachability for nets with trajectories'. In: *LNCS*. PETRI NETS 2024 - 45th International Conference on Theory and Application of Petri Nets and Concurrency. Vol. 14628. Lecture Notes in Computer Science. Genève, Switzerland: Sprincer; Springer Nature Switzerland, 13th June 2024, pp. 244–265. DOI: \url{https://doi.org/10.1007/978-3-031-61433-0_12}. URL: https://inria.hal.science/hal-04711503 (cit. on p. 10).

[14] L. Henry, T. Jéron, N. Markey and V. Roussanaly. 'Distributed Monitoring of Timed Properties'. In: *LNCS24th International Conference, RV 2024, Istanbul, Turkey, October 15–17, 2024, Proceedings*. RV 2024 - 24th International Conference on Runtime Verification. Vol. 15191. Istanbul, Turkey: Springer, 20th Nov. 2024, p. 260. URL: https://hal.science/hal-04701084 (cit. on p. 11).

[15] A. Thébault, L. Hélouët and K. Saiah. 'Modeling subway networks and passenger flows'. In: *OASIcs*. ATMOS 2024 - 24th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems. Vol. 123. Egham, United Kingdom, 2024, pp. 1–20. DOI: 10.4230/OASIcs.ATMOS.2024.17. URL: https://inria.hal.science/hal-04711536 (cit. on p. 12).

Scientific books

[16] N. Bertrand, P. Bouyer, R. Brenguier, A. Carayol, J. Fearnley, H. Gimbert, F. Horn, R. Ibsen-Jensen, N. Markey, B. Monmege, P. Novotny, M. Randour, O. Sankur, S. Schmitz, O. Serre and M. Skomra. *Games on Graphs: From Logic and Automata to Algorithms*. Ed. by N. Fijalkow. 2024, pp. 1–491. DOI: 10.48550/arXiv.2305.10546. URL: https://hal.science/hal-04273394. In press (cit. on p. 13).

Doctoral dissertations and habilitation theses

[17] N. Waldburger. 'Parameterized verification of distributed shared-memory systems'. Université de Rennes, 11th Dec. 2024. URL: https://theses.hal.science/tel-04919634.

Reports & preprints

[18] A. Côme, É. Fabre and L. Hélouët. *A Floyd-Warshall Approach to Value Computation in Markov Decision Processes*. 13th Jan. 2025. URL: https://inria.hal.science/hal-04883133.

[19] O. Sankur, T. Jéron, N. Markey, D. Mentré and R. Noguchi. *Online Test Synthesis From Requirements: Enhancing Reinforcement Learning with Game Theory*. 2024. URL: https://hal.science/hal-04662214 (cit. on p. 11).

[20] A. Thébault, L. Hélouët and K. Harkouken Saiah. *Delay Propagation in Metro Timetables*. 2024. URL: https://inria.hal.science/hal-04711540 (cit. on p. 12).

12.3 Cited publications

[21] 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. 4).

[22] 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. 6).

[23] 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. 6).

[24] 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. 5).

[25] 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. 4).

[26] 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. 7).

[27] 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. 7).

[28] 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. 7).

[29] 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. 7).

[30] 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. 5).

[31] 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. 5).

[32] 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. 7).

[33] 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. 7).

[34] 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. 7).

[35] E. Clarke, D. Long and K. McMillan. 'Compositional model checking'. In: *LICS'89*. IEEE Comp. Soc. Press, 1989, pp. 353–362 (cit. on p. 5).

[36] 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. 5).

[37] 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. 5).

[38] 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. 5).

[39] 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. 6).

[40] 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. 8).

[41] 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. 7).

[42] 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. 7).

[43] 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. 6).

[44] 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. 8).

[45] 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. 6).

[46] 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. 5).

[47] 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. 4).

[48] 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. 7).

[49] 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. 5).

[50] 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. 5).

[51] F. Herbreteau and G. Point. *TChecker*. https://github.com/ticktac-project/tchecker. 2019 (cit. on pp. 4, 5).

[52] 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. 7).

[53] 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. 8).

[54] 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. 5).

[55] 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. 4).

[56] 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. 3).

[57] 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. 6).

[58] 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. 4).

[59] 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. 6).

[60] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014 (cit. on p. 7).

[61] 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. 6).

[62] 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. 5).

[63] O. Sankur and B. Thomas. 'PyLTA: A Verification Tool for Parameterized Distributed Algorithms'. In: *TACAS'23*. Vol. 13994. LNCS. Springer, 2023 (cit. on p. 7).

[64] H. A. Simon. 'Rational Choice and the Structure of the Environment'. In: *Psychological Review* 63.2 (1956), pp. 129–138 (cit. on p. 7).

[65] 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. 7).

[66] 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. 5).