

2025 Activity Report

RESEARCH CENTRE: Inria Centre at Université Grenoble Alpes
IN PARTNERSHIP WITH: Université de Grenoble Alpes

Project-Team

CONVECS

Construction of verified concurrent systems

In collaboration with Laboratoire d'Informatique de Grenoble (LIG)



Project-Team CONVECS

Creation of the Project-Team: 2014 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

- A1.1.2. – Hardware accelerators (GPGPU, FPGA, etc.)
- A1.3.3. – Blockchain
- A1.3.5. – Cloud
- A2.1.1. – Semantics of programming languages
- A2.1.6. – Concurrent programming
- A2.1.7. – Distributed programming
- A2.3.2. – Cyber-physical systems
- A2.5. – Software engineering
 - A2.5.1. – Software Architecture & Design
 - A2.5.4. – Software Maintenance & Evolution
 - A2.5.5. – Software testing
- A4.5.1. – Static analysis
- A4.5.2. – Model-checking
- A5.10.1. – Design
- A6.1.3. – Discrete Modeling (multi-agent, people centered)
- A7.1.1. – Distributed algorithms
- A7.1.3. – Graph algorithms
- A7.2. – Logic in Computer Science
- A8.9. – Performance evaluation
- A9.11. – Generative AI

Other research topics and application domains

- B2. – Digital health
- B5.1. – Factory of the future
- B5.4. – Microelectronics
- B5.6. – Robotic systems
- B6.1.1. – Software engineering
- B6.3.2. – Network protocols
- B6.5. – Information systems
- B6.6. – Embedded systems
- B7.2.1. – Smart vehicles

Contents

Project-Team CONVECS	1
1 Team members, visitors, external collaborators	5
2 Overall objectives	6
2.1 Overview	6
3 Research program	6
3.1 New Formal Languages and their Concurrent Implementations	6
3.2 Parallel and Distributed Verification	7
3.3 Timed, Probabilistic, and Stochastic Extensions	7
3.4 Component-Based Architectures for On-the-Fly Verification	8
3.5 Real-Life Applications and Case Studies	9
4 Application domains	9
5 Latest software developments, platforms, open data	10
5.1 Latest software developments	10
5.1.1 CADP	10
5.1.2 TRAIAN	12
6 New results	12
6.1 New Formal Languages and their Implementations	12
6.1.1 LNT Specification Language	12
6.1.2 Formal Modeling and Analysis of BPMN	13
6.1.3 SYNTAX Compiler Generator	14
6.1.4 Other Language Developments	15
6.2 Parallel and Distributed Verification	15
6.2.1 Cloud Computing	15
6.2.2 Blockchain for Digital Health	16
6.3 Timed, Probabilistic, and Stochastic Extensions	17
6.3.1 Nondeterminism in Interactive Markov Chains	17
6.4 Component-Based Architectures for On-the-Fly Verification	17
6.5 Real-Life Applications and Case Studies	18
6.5.1 Autonomous Car	18
6.5.2 Job-Shop Scheduling	19
6.5.3 High-Performance Data Cache	19
6.5.4 Algorand Consensus Protocol	19
6.5.5 Mobile Robots	20
7 Bilateral contracts and grants with industry	20
7.1 Bilateral grants with industry	20
7.1.1 Euris	20
7.1.2 Public IA	20
8 Partnerships and cooperations	20
8.1 International initiatives	20
8.1.1 Other international collaborations	21
8.2 European initiatives	21
8.2.1 Horizon Europe	21
8.2.2 Other european programs/initiatives	23
8.3 National initiatives	23
8.3.1 PEPR Cloud	23
8.3.2 Other national collaborations	24

8.4	Regional initiatives	24
8.4.1	Région Auvergne-Rhône-Alpes	24
9	Dissemination	24
9.1	Promoting scientific activities	25
9.1.1	Scientific events: organisation	25
9.1.2	Scientific events: selection	25
9.1.3	Journal	26
9.1.4	Software dissemination and internet visibility	26
9.1.5	Invited talks	27
9.1.6	Research administration	27
9.2	Teaching - Supervision - Juries - Educational and pedagogical outreach	28
9.2.1	Teaching	28
9.2.2	Supervision	28
9.2.3	Juries	29
10	Scientific production	29
10.1	Major publications	29
10.2	Publications of the year	30
10.3	Cited publications	31

1 Team members, visitors, external collaborators

Research Scientists

- Radu Mateescu [Team leader, Inria, Senior Researcher, HDR]
- Hubert Garavel [Inria, Senior Researcher]
- Frederic Lang [Inria, Researcher]
- Wendelin Serwe [Inria, Researcher]
- Aline Uwimbabazi [Inria, Starting Research Position, from Aug 2025]

Faculty Member

- Gwen Salaün [UGA, Professor, HDR]

Post-Doctoral Fellows

- Wei Chen [Inria, Post-Doctoral Fellow]
- Aline Uwimbabazi [Schlumberger Foundation, until Jul 2025]

PhD Students

- Zachary Assoumani [Inria]
- Nabil Bouchta [Public IA, CIFRE, from Nov 2025]
- Suraj Gupta [Euris, CIFRE]
- Ahmed Khebbeb [UGA]
- Quentin Nivon [UGA]

Technical Staff

- Abdelilah Mejdoubi [Inria, Engineer, until Mar 2025]

Interns and Apprentices

- Apiram Aruchunarajah [Inria, Intern, from Feb 2025 until Aug 2025]
- David Cremer [FLORALIS, Intern, from Feb 2025 until Jul 2025]
- Amandine Heuls [Inria, Intern, until Jun 2025]

Administrative Assistant

- Myriam Etienne [Inria]

External Collaborator

- Pierre Boullier [retired Inria senior researcher]

2 Overall objectives

2.1 Overview

The CONVECS project-team addresses the rigorous design of concurrent asynchronous systems using formal methods and automated analysis. These systems comprise several activities that execute simultaneously and autonomously (i.e., without the assumption about the existence of a global clock), synchronize, and communicate to accomplish a common task. In computer science, asynchronous concurrency arises typically in hardware, software, and telecommunication systems, but also in parallel and distributed programs.

Asynchronous concurrency is becoming ubiquitous, from the micro-scale of embedded systems (asynchronous logic, networks-on-chip, GALS – *Globally Asynchronous, Locally Synchronous* systems, multi-core processors, etc.) to the macro-scale of grids and cloud computing. In the race for improved performance and lower power consumption, computer manufacturers are moving towards asynchrony. This increases the complexity of the design by introducing nondeterminism, thus requiring a rigorous methodology, based on formal methods assisted by analysis and verification tools.

There exist several approaches to formal verification, such as theorem proving, static analysis, and model checking, with various degrees of automation. When dealing with asynchronous systems involving complex data types, verification methods based on state space exploration (reachability analysis, model checking, equivalence checking, etc.) are today the most successful way to detect design errors that could not be found otherwise. However, these verification methods have several limitations: they are not easily accepted by industry engineers, they do not scale well while the complexity of designs is ever increasing, and they require considerable computing power (both storage capacity and execution speed). These are the challenges that CONVECS seeks to address.

To achieve significant impact in the design and analysis of concurrent asynchronous systems, several research topics must be addressed simultaneously. There is a need for user-friendly, intuitive, yet formal specification languages that will be attractive to designers and engineers. These languages should provide for both functional aspects (as needed by formal verification) and quantitative ones (to enable performance evaluation and architecture exploration). These languages and their associated tools should be smoothly integrated into large-scale design flows. Finally, verification tools should be able to exploit the parallel and distributed computing facilities that are now ubiquitous, from desktop to high-performance computers.

3 Research program

3.1 New Formal Languages and their Concurrent Implementations

We aim at proposing and implementing new formal languages for the specification, implementation, and verification of concurrent systems. In order to provide a complete, coherent methodological framework, two research directions must be addressed:

- *Model-based specifications*: these are operational (i.e., constructive) descriptions of systems, usually expressed in terms of processes that execute concurrently, synchronize together and communicate. Process calculi are typical examples of model-based specification languages. The approach we promote is based on LOTOS NT (LNT for short), a formal specification language that incorporates most constructs stemming from classical programming languages, which eases its acceptance by students and industry engineers. LNT [7] is derived from the ISO standard E-LOTOS (2001), of which it represents the first successful implementation, based on a source-level translation from LNT to the former ISO standard LOTOS (1989). We are working both on the semantic foundations of LNT (enhancing the language with module interfaces and timed/probabilistic/stochastic features, compiling the m among n synchronization, etc.) and on the generation of efficient parallel and distributed code. Once equipped with these features, LNT will enable formally verified asynchronous concurrent designs to be implemented automatically.
- *Property-based specifications*: these are declarative (i.e., non-constructive) descriptions of systems, which express *what* a system should do rather than *how* the system should do it. Temporal logics and μ -calculi are typical examples of property-based specification languages. The natural models

underlying value-passing specification languages, such as LNT, are Labeled Transition Systems (LTSs or simply *graphs*) in which the transitions between states are labeled by actions containing data values exchanged during handshake communications. In order to reason accurately about these LTSs, temporal logics involving data values are necessary. The approach we promote is based on MCL (*Model Checking Language*) [36], which extends the modal μ -calculus with data-handling primitives, fairness operators encoding generalized Büchi automata, and a functional-like language for describing complex transition sequences. We are working both on the semantic foundations of MCL (extending the language with new temporal and hybrid operators, translating these operators into lower-level formalisms, enhancing the type system, etc.) and also on improving the MCL on-the-fly model checking technology (devising new algorithms, enhancing ergonomics by detecting and reporting vacuity, etc.).

We address these two directions simultaneously, yet in a coherent manner, with a particular focus on applicable concurrent code generation and computer-aided verification.

3.2 Parallel and Distributed Verification

Exploiting large-scale high-performance computers is a promising way to augment the capabilities of formal verification. The underlying problems are far from trivial, making the correct design, implementation, fine-tuning, and benchmarking of parallel and distributed verification algorithms long-term and difficult activities. Sequential verification algorithms cannot be reused as such for this task: they are inherently complex, and their existing implementations reflect several years of optimizations and enhancements. To obtain good speedup and scalability, it is necessary to invent new parallel and distributed algorithms rather than to attempt a parallelization of existing sequential ones. We seek to achieve this objective by working along two directions:

- *Rigorous design*: Because of their high complexity, concurrent verification algorithms should themselves be subject to formal modeling and verification, as confirmed by recent trends in the certification of safety-critical applications. To facilitate the development of new parallel and distributed verification algorithms, we promote a rigorous approach based on formal methods and verification. Such algorithms will be first specified formally in LNT, then validated using existing model checking algorithms of the CADP toolbox. Second, parallel or distributed implementations of these algorithms will be generated automatically from the LNT specifications, enabling them to be experimented on large computing infrastructures, such as clusters and grids. As a side-effect, this “bootstrapping” approach would produce new verification tools that can later be used to self-verify their own design.
- *Performance optimization*: In devising parallel and distributed verification algorithms, particular care must be taken to optimize performance. These algorithms will face concurrency issues at several levels: grids of heterogeneous clusters (architecture-independence of data, dynamic load balancing), clusters of homogeneous machines connected by a network (message-passing communication, detection of stable states), and multi-core machines (shared-memory communication, thread synchronization). We will seek to exploit the results achieved in the parallel and distributed computing field to improve performance when using thousands of machines by reducing the number of connections and the messages exchanged between the cooperating processes carrying out the verification task. Another important issue is the generalization of existing LTS representations (explicit, implicit, distributed) in order to make them fully interoperable, such that compilers and verification tools can handle these models transparently.

3.3 Timed, Probabilistic, and Stochastic Extensions

Concurrent systems can be analyzed from a *qualitative* point of view, to check whether certain properties of interest (e.g., safety, liveness, fairness, etc.) are satisfied. This is the role of functional verification, which produces Boolean (yes/no) verdicts. However, it is often useful to analyze such systems from a *quantitative* point of view, to answer non-functional questions regarding performance over the long run, response time, throughput, latency, failure probability, etc. Such questions, which call for numerical (rather than binary) answers, are essential when studying the performance and dependability (e.g., availability, reliability, etc.) of complex systems.

Traditionally, qualitative and quantitative analyzes are performed separately, using different modeling languages and different software tools, often by distinct persons. Unifying these separate processes to form a seamless design flow with common modeling languages and analysis tools is therefore desirable, for both scientific and economic reasons. Technically, the existing modeling languages for concurrent systems need to be enriched with new features for describing quantitative aspects, such as probabilities, weights, and time. Such extensions have been well-studied and, for each of these directions, there exist various kinds of automata, e.g., discrete-time Markov chains for probabilities, weighted automata for weights, timed automata for hard real-time, continuous-time Markov chains for soft real-time with exponential distributions, etc. Nowadays, the next scientific challenge is to combine these individual extensions altogether to provide even more expressive models suitable for advanced applications.

Many such combinations have been proposed in the literature, and there is a large amount of models adding probabilities, weights, and/or time. However, an unfortunate consequence of this diversity is the confuse landscape of software tools supporting such models. Dozens of tools have been developed to implement theoretical ideas about probabilities, weights, and time in concurrent systems. Unfortunately, these tools do not interoperate smoothly, due both to incompatibilities in the underlying semantic models and to the lack of common exchange formats.

To address these issues, CONVECS follows two research directions:

- *Unifying the semantic models.* Firstly, we will perform a systematic survey of the existing semantic models in order to distinguish between their essential and non-essential characteristics, the goal being to propose a unified semantic model that is compatible with process calculi techniques for specifying and verifying concurrent systems. There are already proposals for unification either theoretical (e.g., Markov automata) or practical (e.g., PRISM and MODEST modeling languages), but these languages focus on quantitative aspects and do not provide high-level control structures and data handling features (as LNT does, for instance). Work is therefore needed to unify process calculi and quantitative models, still retaining the benefits of both worlds.
- *Increasing the interoperability of analysis tools.* Secondly, we will seek to enhance the interoperability of existing tools for timed, probabilistic, and stochastic systems. Based on scientific exchanges with developers of advanced tools for quantitative analysis, we plan to evolve the CADP toolbox as follows: extending its perimeter of functional verification with quantitative aspects; enabling deeper connections with external analysis components for probabilistic, stochastic, and timed models; and introducing architectural principles for the design and integration of future tools, our long-term goal being the construction of a European collaborative platform encompassing both functional and non-functional analyzes.

3.4 Component-Based Architectures for On-the-Fly Verification

On-the-fly verification fights against state explosion by enabling an incremental, demand-driven exploration of LTSs, thus avoiding their entire construction prior to verification. In this approach, LTS models are handled implicitly by means of their *post* function, which computes the transitions going out of given states and thus serves as a basis for any forward exploration algorithm. On-the-fly verification tools are complex software artifacts, which must be designed as modularly as possible to enhance their robustness, reduce their development effort, and facilitate their evolution. To achieve such a modular framework, we undertake research in several directions:

- *New interfaces for on-the-fly LTS manipulation.* The current application programming interface (API) for on-the-fly graph manipulation, named OPEN/CAESAR [25], provides an “opaque” representation of states and actions (transitions labels): states are represented as memory areas of fixed size and actions are character strings. Although appropriate to the pure process algebraic setting, this representation must be generalized to provide additional information supporting an efficient construction of advanced verification features, such as: handling of the types, functions, data values, and parallel structure of the source program under verification, independence of transitions in the LTS, quantitative (timed/probabilistic/stochastic) information, etc.
- *Compositional framework for on-the-fly LTS analysis.* On-the-fly model checkers and equivalence checkers usually perform several operations on graph models (LTSs, Boolean graphs, etc.), such as

exploration, parallel composition, partial order reduction, encoding of model checking and equivalence checking in terms of Boolean equation systems, resolution and diagnostic generation for Boolean equation systems, etc. To facilitate the design, implementation, and usage of these functionalities, it is necessary to encapsulate them in software components that could be freely combined and replaced. Such components would act as graph transformers, that would execute (on a sequential machine) in a way similar to coroutines and to the composition of lazy functions in functional programming languages. Besides its obvious benefits in modularity, such a component-based architecture will also make it possible to take advantage of multi-core processors.

- *New generic components for on-the-fly verification.* The quest for new on-the-fly components for LTS analysis must be pursued, with the goal of obtaining a rich catalog of interoperable components serving as building blocks for new analysis features. A long-term goal of this approach is to provide an increasingly large catalog of interoperable components covering all verification and analysis functionalities that appear to be useful in practice. It is worth noticing that some components can be very complex pieces of software (e.g., the encapsulation of an on-the-fly model checker for a rich temporal logic). Ideally, it should be possible to build a novel verification or analysis tool by assembling on-the-fly graph manipulation components taken from the catalog. This would provide a flexible means of building new verification and analysis tools by reusing generic, interoperable model manipulation components.

3.5 Real-Life Applications and Case Studies

We believe that theoretical studies and tool developments must be confronted with significant case studies to assess their applicability and to identify new research directions. Therefore, we seek to apply our languages, models, and tools for specifying and verifying formally real-life applications, often in the context of industrial collaborations.

4 Application domains

The theoretical framework we use (automata, process algebras, bisimulations, temporal logics, etc.) and the software tools we develop are general enough to fit the needs of many application domains. They are applicable to virtually any system or protocol that consists of distributed agents communicating by asynchronous messages. The list of recent case studies performed with the CADP toolbox (see in particular § 6.5) illustrates the diversity of applications:

- *Bioinformatics:* genetic regulatory networks, nutritional stress response, metabolic pathways,
- *Component-based systems:* Web services, peer-to-peer networks,
- *Cloud computing:* self-deployment protocols, dynamic reconfiguration protocols,
- *Databases:* transaction protocols, distributed knowledge bases, stock management,
- *Distributed systems:* virtual shared memory, dynamic reconfiguration algorithms, fault tolerance algorithms, multi-agent systems,
- *Embedded systems:* air traffic control, autonomous vehicles, avionics systems, train supervision systems, medical devices,
- *Enterprise systems:* business processes, information systems, manufacturing,
- *Fog and IoT:* stateful IoT applications in the fog, industrial IoT,
- *Hardware architectures:* multiprocessor architectures, systems on chip, cache coherency protocols, hardware/software codesign,
- *Human-machine interaction:* graphical interfaces, biomedical data visualization, plasticity,

- *Security protocols*: authentication, blockchain, electronic transactions, cryptographic key distribution,
- *Telecommunications*: high-speed networks, network management, mobile telephony, feature interaction detection.

5 Latest software developments, platforms, open data

5.1 Latest software developments

5.1.1 CADP

Name: Construction and Analysis of Distributed Processes

Keywords: Formal methods, Verification

Functional Description: CADP (*Construction and Analysis of Distributed Processes* – formerly known as *CAESAR/ALDEBARAN Development Package*) [6] is a toolbox for protocols and distributed systems engineering.

In this toolbox, we develop and maintain the following tools:

- CAESAR.ADT [24] is a compiler that translates LOTOS abstract data types into C types and C functions. The translation involves pattern-matching compiling techniques and automatic recognition of usual types (integers, enumerations, tuples, etc.), which are implemented optimally.
- CAESAR [30, 29] is a compiler that translates LOTOS processes into either C code (for rapid prototyping and testing purposes) or finite graphs (for verification purposes). The translation is done using several intermediate steps, among which the construction of a Petri net extended with typed variables, data handling features, and atomic transitions.
- OPEN/CAESAR [25] is a generic software environment for developing tools that explore graphs on the fly (for instance, simulation, verification, and test generation tools). Such tools can be developed independently of any particular high level language. In this respect, OPEN/CAESAR plays a central role in CADP by connecting language-oriented tools with model-oriented tools. OPEN/CAESAR consists of a set of 16 code libraries with their programming interfaces, such as:
 - CAESAR_GRAPH, which provides the programming interface for graph exploration,
 - CAESAR_HASH, which contains several hash functions,
 - CAESAR_SOLVE, which resolves Boolean equation systems on the fly,
 - CAESAR_STACK, which implements stacks for depth-first search exploration, and
 - CAESAR_TABLE, which handles tables of states, transitions, labels, etc.

A number of on-the-fly analysis tools have been developed within the OPEN/CAESAR environment, among which:

- BISIMULATOR, which checks bisimulation equivalences and preorders,
- CUNCTATOR, which performs steady-state simulation of continuous-time Markov chains,
- DETERMINATOR, which eliminates stochastic nondeterminism in normal, probabilistic, or stochastic systems,
- DISTRIBUTOR, which generates the graph of reachable states using several machines,
- EVALUATOR, which evaluates MCL formulas,
- EXECUTOR, which performs random execution,
- EXHIBITOR, which searches for execution sequences matching a given regular expression,
- GENERATOR, which constructs the graph of reachable states,
- PROJECTOR, which computes abstractions of communicating systems,
- REDUCTOR, which constructs and minimizes the graph of reachable states modulo various equivalence relations,
- SIMULATOR, XSIMULATOR, and OCIS, which enable interactive simulation, and

- TERMINATOR, which searches for deadlock states.
- BCG (*Binary Coded Graphs*) is both a file format for storing very large graphs on disk (using efficient compression techniques) and a software environment for handling this format. BCG also plays a key role in CADP as many tools rely on this format for their inputs/outputs. The BCG environment consists of various libraries with their programming interfaces, and of several tools, such as:
 - BCG_CMP, which compares two graphs,
 - BCG_DRAW, which builds a two-dimensional view of a graph,
 - BCG_EDIT, which allows the graph layout produced by BCG_DRAW to be modified interactively,
 - BCG_GRAPH, which generates various forms of practically useful graphs,
 - BCG_INFO, which displays various statistical information about a graph,
 - BCG_IO, which performs conversions between BCG and many other graph formats,
 - BCG_LABELS, which hides and/or renames (using regular expressions) the transition labels of a graph,
 - BCG_MIN, which minimizes a graph modulo strong or branching equivalences (and can also deal with probabilistic and stochastic systems),
 - BCG_STEADY, which performs steady-state numerical analysis of (extended) continuous-time Markov chains,
 - BCG_TRANSIENT, which performs transient numerical analysis of (extended) continuous-time Markov chains, and
 - XTL (*eXecutable Temporal Language*), which is a high level, functional language for programming exploration algorithms on BCG graphs. XTL provides primitives to handle states, transitions, labels, *successor* and *predecessor* functions, etc. For instance, one can define recursive functions on sets of states, which allow evaluation and diagnostic generation fixed point algorithms for usual temporal logics (such as HML [31], CTL[22], ACTL[23], etc.) to be defined in XTL.
- PBG (*Partitioned BCG Graph*) is a file format implementing the theoretical concept of *Partitioned LTS* [28] and providing a unified access to a graph partitioned in fragments distributed over a set of remote machines, possibly located in different countries. The PBG format is supported by several tools, such as:
 - PBG_CP, PBG_MV, and PBG_RM, which facilitate standard operations (copying, moving, and removing) on PBG files, maintaining consistency during these operations,
 - PBG_MERGE (formerly known as BCG_MERGE), which transforms a distributed graph into a monolithic one represented in BCG format,
 - PBG_INFO, which displays various statistical information about a distributed graph.
- The connection between explicit models (such as BCG graphs) and implicit models (explored on the fly) is ensured by OPEN/CAESAR-compliant compilers, e.g.:
 - BCG_OPEN, for models represented as BCG graphs,
 - CAESAR.OPEN, for models expressed as LOTOS descriptions,
 - EXP.OPEN, for models expressed as communicating automata,
 - FSP.OPEN, for models expressed as FSP [34] descriptions,
 - LNT.OPEN, for models expressed as LNT descriptions, and
 - SEQ.OPEN, for models represented as sets of execution traces.

The CADP toolbox also includes TGV (*Test Generation based on Verification*), which has been developed by the VERIMAG laboratory (Grenoble) and Inria Rennes – Bretagne-Atlantique.

The CADP tools are well-integrated and can be accessed easily using either the EUCALYPTUS graphical interface or the SVL [26] scripting language. Both EUCALYPTUS and SVL provide users with an easy and uniform access to the CADP tools by performing file format conversions automatically whenever needed and by supplying appropriate command-line options as the tools are invoked.

URL: <http://cadp.inria.fr/>

Contact: Hubert Garavel

Participants: Hubert Garavel, Frederic Lang, Radu Mateescu, Wendelin Serwe

5.1.2 TRAIAN

Keywords: Compilation, LOTOS NT

Functional Description: TRAIAN is a compiler for translating LOTOS NT descriptions into C programs, which will be used for simulation, rapid prototyping, verification, and testing.

The current version of TRAIAN, which handles LOTOS NT types and functions only, has useful applications in compiler construction [27], being used in all recent compilers developed by CONVECS.

URL: <http://convecs.inria.fr/software/traian/>

Contact: Hubert Garavel

Participants: Hubert Garavel, Frederic Lang, Wendelin Serwe

6 New results

6.1 New Formal Languages and their Implementations

6.1.1 LNT Specification Language

Participants: Hubert Garavel (*correspondent*), Frédéric Lang, Wendelin Serwe.

LNT [7] [21] is a next-generation formal description language for asynchronous concurrent systems. The design of LNT at CONVECS is the continuation of the efforts undertaken in the 80s to define sound languages for concurrency theory and, indeed, LNT is derived from the ISO standards LOTOS (1989) and E-LOTOS (2001). In a nutshell, LNT attempts to combine the best features of imperative programming languages, functional languages, and value-passing process calculi.

LNT is not a frozen language: its definition started in 2005, as part of an industrial project. Since 2010, LNT has been systematically used by CONVECS for numerous case studies (many of which being industrial applications — see § 6.5). LNT is also used as a back-end by other research teams who implement various languages by translation to LNT. It is taught in university courses, e.g., at University Grenoble Alpes and ENSIMAG, where it is positively accepted by students and industry engineers. Based on the feedback acquired by CONVECS, LNT is continuously improved.

LOTOS NT is a language predecessor of LNT, equipped with the TRAIAN compiler, which is used for the construction of most CADP compilers and translators. In 2025, the unification of LNT and LOTOS NT was achieved, with the removal of all references to LOTOS NT in tools and documentation. As a consequence, from January 2025, newer versions of TRAIAN have been distributed as part of the monthly releases of CADP. In parallel, the releases of TRAIAN have continued independently of CADP on a less frequent schedule, with a single consolidated version TRAIAN 3.17 released in 2025, which improves the static semantics checking (enhanced warning and error messages, more precise data-flow analysis on array types and “case” patterns, detection of irrelevant “require” and “ensure” clauses, detection of constant loop conditions).

In 2025, the LNT language was enhanced in several respects:

- A new iterative construct “for ... until” (with breakable and unbreakable forms), useful to scan the domain of enumerated and range types, was introduced in both LNT functions and processes.

- LNT was enriched with virtual processes, types, and channels (all defined using the “!virtual” pragma), which enables one to specify generic LNT modules parameterized by types, functions, processes, and/or channels.
- Two new instructions “+=” and “-=” for incrementing and decrementing variables or array elements (for numeric types and all types equipped with binary functions named “+” or “-”) have been added.
- The syntax of LNT was made stricter, so as to ban ambiguous combinations of prefix unary operators and postfix notations, such as type conversions (“V of T”), field selections (“V.X”), field updates (“V.{X -> ...}”), and array accesses (“V1 [V2]”).
- The “..” keyword (used in the definition of range and array types) was deprecated and replaced by the “...” keyword (used in lists and sets). The former notation “!X” for “in out” parameters was deprecated and replaced by a simpler notation “X?”.
- A new pragma “!library” was introduced to suppress the warnings emitted about functions and processes never called and/or types and channels never employed. Pragmas “!implementedby” can now be attached to “with get” and “with set” clauses to fix the names of the corresponding selector and updater functions.

Additionally, the predefined libraries of LNT have been enriched with over 70 new functions to increase user-friendliness (“min” and “max” functions for all types, “inf” and “sup” for list and set types, “+” and “-” for range types, etc.).

The LNT documentation was updated accordingly and the LNT2LOTOS Reference Manual was enriched with 14 pages of examples taken from courses and exams given at ENSIMAG and UGA.

6.1.2 Formal Modeling and Analysis of BPMN

Participants: David Cremer, Frédéric Lang, Quentin Nivon, Gwen Salaün (*correspondent*).

Modelling and designing business processes has become a crucial activity for companies in the last 20 years. As a consequence, multiple workflow modelling notations were proposed. BPMN (*Business Process Modelling Notation*) is one of them and is now considered as the de facto standard for process modelling.

In 2025, we contributed to the generation, verification, and maintenance of BPMN processes along two directions.

- One existing challenge is to enhance the design of BPMN processes with formal methods in order to avoid erroneous descriptions and to ensure correct process executions.

Since classic verification techniques require a certain level of expertise, we proposed an approach accessible to any kind of users, either novices or experts, as one can describe both processes and their functional requirements in natural language. We implemented the approach in the GIVUP (Generation and Verification of Underspecified Processes) tool, which takes as input a textual description of a process in natural language and automatically generates a corresponding BPMN process. GIVUP first uses an LLM (Large Language Model) to extract task dependencies and additional information from the description, and then manipulates these dependencies to generate a single BPMN process. The tool can also take as input a textual description of a functional property that must be preserved, and checks whether the process satisfies the property. This is achieved by transforming both the BPMN process into a language understandable by model checkers, and the textual property into LTL (Linear Temporal Logic), and finally by verifying the property on the process model using model checking. If the property is violated, the diagnostic (visualized as classic counterexample, set of all counterexamples, or coloured BPMN process) can be used to refine the process description. GIVUP was applied to a large set of examples for evaluation purposes. This work led to a publication in an international conference [14]. A detailed account of the work carried out on the generation, analysis, and optimization of BPMN processes is available in Quentin Nivon’s PhD thesis [37].

- Maintaining consistency between business process models and their textual descriptions is critical for operational clarity, compliance, and communication. However, as process models evolve, updating documentation remains costly and error-prone.

In collaboration with Benjamin Dalmas (iGrafx), we proposed an automated approach that treats process-text consistency as an edit-based synchronization problem. The approach combines graph-based model differencing with prompt-guided editing by LLMs. For balanced acyclic BPMN processes, we introduce the Longest Common Execution Subsequence (LCES) algorithm to isolate shared control-flow, while for processes containing loops or unbalanced gateways, we employ a custom beam-search heuristic that explores plausible sequences of insertions, deletions, and refinements. Our experiments indicate that this approach can produce text that remains semantically aligned with updated processes and stylistically consistent with original descriptions. This work led to a publication in an international conference [15].

6.1.3 SYNTAX Compiler Generator

Participants: Pierre Boullier, Hubert Garavel (*correspondent*), Frédéric Lang, Wendelin Serwe.

SYNTAX is a compiler-generation tool that generates lexical analyzers (scanners) and syntactic analyzers (parsers) for deterministic and nondeterministic context-free grammars. Developed since 1970, SYNTAX is probably the oldest software of INRIA that is still actively used and maintained. In particular, SYNTAX serves to produce the front-end part of most compilers of CADP, including TRAIAN.

Since the closing of the INRIA Gforge in 2021, the SYNTAX code is hosted on the RENATER SVN repository.

In 2025, the development of SYNTAX has been active, with 661 commits. Significant progress was made in the three following directions.

Enhancements to SYNTAX core tools (“trunk”). In addition to fixing four bugs, the standard library of SYNTAX “libsx.a” was enhanced as follows.

The “sxcommon.h” include was upgraded by porting it to 64-bit Windows, removing Gcc-specific macros, and taking into account recent cross-compilers. The memory manager “sxmem_mgr” was simplified by removing specific code for obsolete architectures (DEC Alpha, SGI, etc.). The string manager “sxstr_mgr” was updated with the new list of reserved keywords of the C23 version of the C language. The TABLES_C processor was simplified by removing the PARSACT macro-definition. The C code produced by this processor now emits warnings if it is compiled with deprecated options.

The “sxmake” script used to compile SYNTAX was enhanced to invoke the advanced static-analysis and runtime-checks available with Gcc on Linux. The script can now compile all SYNTAX extensions using a single command, produces more readable logs, and emits clearer messages. The documentation has also been enhanced.

Demo examples provided with SYNTAX. The SIMPRO demo was updated to follow the evolutions of the LNT language and to no longer use the “-DSEMACT” option nor deprecated Gcc options.

The FORTRAN-77 compiler front-end was upgraded to handle a larger class of FORTRAN extensions, such as CR characters not followed by LF characters, RETURN statements in MAIN programs, out-of-order DATA directives, double- and quadruple-precision constants, as well as binary, octal, and hexadecimal constants. New FORTRAN tests have been added and the “run-test” script was significantly enhanced.

Taking inspiration from the work done by Larisa Safina (EVREF project-team), we started extending the FORTRAN grammar with attributes and semantic actions to generate an abstract syntax tree in JSON format. The “sxml” library was augmented with new functions and a new library “sxjson” was created. The JSON code produced is now automatically indented using mainstream JSON pretty-printers.

Enhancements to SYNTAX additional tools (“extensions”). So far, SYNTAX has been divided into three parts: the main processors (“trunk”) for deterministic parsing of computer languages, the auxiliary processors (“extensions”) for ambiguous parsing of natural languages, and a grey zone (“oldies”) of processors that had not been maintained for years.

In 2025, a major reorganization took place to remove the grey zone. Each “oldie” processor was carefully scrutinized and either moved to “extensions” or definitively archived in “outdated/deleted” if one could not recompile it. Consequently, SYNTAX now only has two parts, “trunk” and “extensions”.

The directory structure of “extension” processors was simplified, from a tree-like to a flat one. A few legacy sub-directories were removed and new include files and modules were created.

Many changes (removal of dead code and useless variables, fixes for type errors, function prototypes, and string buffer overflows, etc.) were brought to ensure that all “extensions” processors of SYNTAX now compile without errors or warnings using the most recent versions of Gcc and Clang for C23.

6.1.4 Other Language Developments

Participants: Hubert Garavel, Frédéric Lang, Radu Mateescu.

In 2025, in addition to various bug fixes in EVALUATOR (option “-bes”) and SVL, various languages and compilers of CADP have been improved as follows:

- The XTL_EXPAND preprocessor, common to the MCL and XTL languages, now supports single-line comments beginning with “--”, in addition to the already supported multi-line comments delimited by “(*/” and “*/)”. The “standard.mcl” library of MCL was improved by introducing eight macro-definitions of common usage that remove the need for writing lower-level μ -calculus formulas.
- The MCL_EXPAND preprocessor was enhanced to report data variables defined but not used in MCL formulas and to reduce the size of formulas (by factoring common prefixes of regular formulas occurring in modalities), which in some cases reduced the verification time by up to 37%.
- The three versions 3, 4, 5 of the EVALUATOR model checker have been merged into a single one, and the source code of EVALUATOR, MCL_EXPAND, and XTL_EXPAND, as well as the C code they produce, were ported to C23 (the latest revision of the C language published in October 2024).
- The syntax of the SVL language was enhanced by extending the “|=” operator to evaluate properties stored in MCL and XTL files, by introducing three new keywords “end abstraction”, “end generation”, and “end reduction” for better structuring, and by simplifying the language following the unification of the three versions of EVALUATOR into a single one.
- The SVL compiler was improved to run SVL scripts as full-fledge programs from the command line, to distinguish more clearly between errors and warnings reported by the various CADP tools invoked, and to directly display the messages emitted by the last CADP command that failed.

6.2 Parallel and Distributed Verification

6.2.1 Cloud Computing

Participants: Ahmed Khebbab, Gwen Salaün (*correspondent*).

Cloud computing becomes increasingly complex due to the emergence of new computing infrastructures (e.g., edge computing or the cloud-edge-IoT computing continuum), which involve diverse and heterogeneous resources, geographical distribution, and increased requirements for dynamicity, security, and energy consumption.

In 2025, in the context of the TARANIS project (see § 8.3.1), we contributed to the cloud computing domain as follows.

TOSCA workflows TOSCA is a textual specification language for modelling cloud application topologies and orchestration workflows, such as deployment and undeployment plans. Purely textual workflow descriptions lack visual support and are prone to specification errors that can affect correctness and reliability. In collaboration with Philippe Merle (SPIRALS project-team), we devised a systematic transformation of TOSCA workflows into BPMN to enable visualisation and to support automated formal verification of functional, architectural, and other application-specific properties. This transformation has been implemented as part of a scalable toolchain. An article presenting this work has been submitted to an international journal.

Kubernetes platform Kubernetes is an open source orchestration platform for automating the configuration, deployment, scaling, and management of containerized applications. The platform is widely used in organizations to run distributed applications and services at scale. In collaboration with Philippe Merle, we considered the improving of Kubernetes in two respects.

Kubernetes orchestrates cloud and microservice deployments, but does not explicitly model dependencies between deployments that rely on each other. Reconfigurations performed without dependency awareness can lead to failures, inefficient recoveries, or unstable behaviour despite Kubernetes self-healing mechanisms. We proposed an approach to model deployments and their dependencies as a dependency graph and to generate coordinated reconfiguration plans from high-level declarative intents using the CESR interpreted language, currently implemented in the Cestrum framework.

We also investigated the usage of formal methods for verifying the correctness of the Kubernetes scheduling (optimizing the placements). The pods (groups of containers with a shared storage and network resources) are scheduled in the order they are given to the scheduler, which can lead to placement conflicts, resulting in some pods being left unschedulable. We proposed a novel solution leveraging SAT solvers and Petri nets for modeling and verifying the feasibility of scheduling pods with an account for their placement constraints. This solution, which allows one to guide the Kubernetes scheduler to achieve a placement in real time, is currently being implemented in the Polyanthum prototype framework.

Infrastructure-as-Code IaC (Infrastructure-as-Code) allows managing infrastructure through different languages and technologies, leading to a diverse and fragmented landscape of IaC languages and approaches. The absence of a clear definition, taxonomy, and comparison framework makes it difficult to understand, classify, and reason about the properties of IaC languages.

In collaboration with Quentin Guilloteau (AVALON project-team), Eloi Perdereau (STACK project-team), Jolan Philippe (LIFO, Orléans), Hélène Coullon (STACK project-team), and Philippe Merle, we elaborated a comprehensive survey to propose a detailed taxonomy and systematic analysis of IaC languages to clarify their characteristics and facilitate their comparison.

6.2.2 Blockchain for Digital Health

Participants: Suraj Gupta, Frédéric Lang, Gwen Salaün (*correspondent*).

Electronic Health Records (EHRs) are essential for modern healthcare technology, having a critical role in various applications (clinical decision support systems, telehealth platforms, population health management tools, patient portals, research platforms for medical innovation). Given their sensitive nature, EHRs are subject to interoperability, security, and efficiency constraints.

In 2025, in collaboration with Umar Ozeer (Euris, see § 7.1.1), we proposed FaSTr, a blockchain-based solution that aims to make the process of interacting with EHRs fast, secure, and transparent by providing services. FaSTr integrates with an EHR storage system and provides secure API endpoints used by authorized applications to access the EHR data. FaSTr handles multiple requests from applications for EHRs in near-real time while maintaining high availability and robustness, enables strict access control policies for accessing EHRs, integrates in a standardized way with EHR storage systems, and provides immutable logging of changes in access policies and all interactions with EHRs. A prototype implementation of FaSTr was developed and experimented to evaluate its latency and throughput, demonstrating that the network scales well with an increasing ledger size. This work led to a publication in an international conference [11].

6.3 Timed, Probabilistic, and Stochastic Extensions

6.3.1 Nondeterminism in Interactive Markov Chains

Participants: Hubert Garavel.

Modeling and analyzing complex systems that combine functional behaviour and quantitative aspects poses a difficult challenge, namely the proper integration of process calculi and (discrete-time or continuous-time) Markov chains. Mainstream process calculi are inherently non-deterministic (e.g., due to their choice operators and to the interleaving semantics of their parallel composition operators), whereas the concept of nondeterminism is absent in Markov chains. Thus, it is not straightforward to define conservative extensions of process calculi with Markov chains.

In 2025, in collaboration with Holger Hermanns (Saarland University, Germany), we investigated three related issues:

- Two types of state-transition models have been proposed for the formal description of continuous-time stochastic systems, in which delays are governed by exponential distributions of known rate parameters. In the first type, each transition consists of an event and a rate glued together, while in the second type, illustrated by the IMCs (Interactive Markov Chains) [32] each transition carries either an event or a rate. This raises compatibility, as well as modeling choice issues, since both types of models are theoretically different.
- The next issues concern the possibility of automated conversions between both types of models. Specifically, we defined a translation function that converts models of the first type into models of the second type. We studied how faithful this translation is with respect to crucial properties such as deadlocks, determinism, and the preservation of steady-state and transient probabilities.
- Finally, we studied the issues related to the presence of nondeterminism in the models of the second type produced by our translation function. We analyzed the reasons why such nondeterminism appears and discussed how to cope with it or to eliminate it using successive transformations of the model.

We investigated these issues on a concrete case study, the “Erlangen mainframe”, for which models of the first type already exist, and for which we developed a new model of the second type in LNT. This work was published as a book chapter [16].

6.4 Component-Based Architectures for On-the-Fly Verification

Participants: Hubert Garavel, Frédéric Lang, Radu Mateescu, Abdelilah Mejdoubi, Wendelin Serwe.

In 2025, in addition to various bug fixes, several enhancements have been brought to the CADP tools related to compositional and on-the-fly verification:

- The performance of the EXP2C tool for networks of communicating automata in the EXP language was significantly enhanced for composition expressions containing priority operators: the CPU time is now quadratic (rather than cubic) in the number of labels. For instance, on a network with over 10,000 distinct labels, EXP2C did not finish within more than an hour, but now terminates in 20 seconds.
- BCG_MIN and BCG_CMP now provide SVL with detailed information about the branching factor when these tools are halted by the operating system due to memory exhaustion. All BCG tools, their code libraries, and their related tools have been updated to comply with C23.
- The OPEN/CAESAR environment for on-the-fly verification was enhanced by introducing new data types (mostly representing pointers to functions) in the programming interfaces. All OPEN/CAESAR-compliant compilers (BCG_OPEN, CAESAR, EXP.OPEN, and SEQ.OPEN) and OPEN/CAESAR

application tools (e.g., CUNCTATOR, DETERMINATOR, XSIMULATOR, etc.), as well as their documentation, have been updated to comply with C23.

Additionally, in all CADP tools having an option “-root”, this option was renamed to “-main”. The windows of the EUCALYPTUS graphical user interface can now be resized freely and the command-line options of the TGV test generation tool were made more user-friendly. CADP can now be downloaded, in addition to FTP, also with HTTP or HTTPS.

All demos have been revised to follow the evolution of CADP languages (virtual types and incrementation/decrementation operations of LNT, new high-level macro-definitions of the MCL standard library, “abstraction” operator of SVL). The two versions of the alternating-bit protocol, demo_01 and demo_02, have been merged into a single one, and demo_02, which was the last CADP demo expressed in LOTOS, was translated to LNT.

6.5 Real-Life Applications and Case Studies

6.5.1 Autonomous Car

Participants: Wei Chen, Jean-Baptiste Horel, Radu Mateescu (*correspondent*), Wendelin Serwe, Aline Uwimbabazi.

Devising scenarios for testing autonomous vehicles (AV) is still a challenging task that requires a tradeoff between cost and achieved coverage of the considered operational design domain. Often scenarios are derived from accident statistics and real traffic datasets. Expertise knowledge for the tasks of selecting the scenarios for testing and/or simulation is highly required. This task is carried out mostly manually, and would benefit from a precise method to assess scenario relevance and compare scenarios.

In collaboration with Lina Marso (Polytechnique Montréal, Canada), Christian Laugier, and Alessandro Renzaglia (CHROMA project-team), we proposed an automatic approach to generate a large number of relevant critical scenarios for autonomous driving simulators. The approach relies on the generation of behavioral conformance tests using the TESTOR tool [35], from an LNT model (specifying the ground truth configuration with the range of vehicle behaviors) and a test purpose (specifying the critical feature under analysis). The obtained test cases (which cover all possible executions exercising a given feature) are automatically translated into the inputs of autonomous driving simulators.

In 2025, in the framework of the A-IQ Ready project (see § 8.2.1), we continued this line of work as follows:

- We pursued the formal modelling and simulation of pedestrian behaviours in shared spaces, focusing on the interactions between pedestrians and AVs. We developed an LNT model of pedestrian behaviour, integrating perception zones, attention mechanisms, and a density-dependent personal space inspired by existing behavioural models [38]. The model combines driving forces toward destinations with social interaction forces between pedestrians, allowing realistic adaptation of trajectories in multi-agent scenarios. Several simplifications were introduced to make the model suitable for formal verification, including a grid-based environment and abstracted physical constraints. The LNT model provides a basis for verifying the safety of interactions between pedestrians and an AV, as well as generating scenarios for testing AVs in complex situations.
- We continued our work on a model-based methodology to compare scenarios using quantitative measures computed from a formal model of an AV in its environment and a set of user-defined interesting event sequences representing evaluation criteria. Quantitative measures are computed using two different approaches: probabilistic model checking of temporal logic properties and conformance testing guided by test purposes—both temporal properties and test purposes being derived from the considered event sequences. This methodology facilitates the selection of the scenarios having the best tradeoff between coverage and overall testing cost (both for simulation and field testing). We applied the methodology to compare variations of five scenarios, derived from frequent situations in accident statistics, using several evaluation criteria (occurrence of collision, successful arrival, duration). This work led to a publication in an international conference [12] and to an article submitted to an international journal.

6.5.2 Job-Shop Scheduling

Participants: Radu Mateescu, Wendelin Serwe (*correspondent*), Aline Uwimbabazi.

The job-shop scheduling problem is a well-known NP-hard scheduling problem, where a set of jobs (consisting of a sequence of tasks) have to be executed on a set of machines, each task specifying its duration and the required machine.

In 2025, in the framework of the A-IQ Ready project (see § 8.2.1), we pursued our work on the Job-Shop Scheduling Problem to compute schedules for autonomous robots using the state-space exploration algorithms of CADP. We continued experimenting several encodings of the problem in the LNT language, with the goal of developing efficient models that can provide better solutions than those found in some benchmark instances or found using reinforcement learning. Our models can be used to study the Job-Shop Scheduling Problem because we were able to generate the complete Labelled Transition System (LTS) of a given instance, which allowed us to find all possible solutions. This work led to a book chapter that was accepted for publication.

6.5.3 High-Performance Data Cache

Participants: Zachary Assoumani, Radu Mateescu, Wendelin Serwe (*correspondent*).

To reduce costs in hardware design, it is crucial to spot any unwanted behaviours early in the design process of complex architectures, starting with the informal specifications, which are particularly prone to ambiguities and oversights. Formal methods are suitable vehicles to describe, simulate, and also verify specifications of architectures.

In 2025, in collaboration with César Fugueta (MADMAX project-team), we considered the formal modeling and analysis of the High-Performance Data Cache (HPDcache) [41], a non-blocking L1 data cache for RISC-V cores and accelerators. Starting from its informal specification, we produced a formal model of the HPDcache in LNT. We also formally expressed the memory consistency rules of the RISC-V specification as MCL formulas. Using the CADP tools supporting LNT and MCL, we uncovered a possible violation of the memory consistency rules by the informal specification of the HPDcache. We created an *issue* in the official HPDcache repository, which has been fixed (the SystemVerilog code was not affected). This work led to an extended abstract presented as poster at an international conference [19].

6.5.4 Algorand Consensus Protocol

Participants: Hubert Garavel.

A blockchain is a distributed, tamper-proof ledger system that permanently records transactions across a network of possibly untrusted nodes. The ledger maintains a cryptographically linked list of committed blocks, each one containing a reference to the previous block, forming an immutable chain of records. In its pure form (called public or permissionless blockchain), any node can join or leave the system at any time, with agreement on transaction history being reached via a consensus protocol instead of a centralized trusted party. Algorand is a scalable and secure permissionless blockchain that achieves proof-of-stake consensus via cryptographic self-sortition and binary Byzantine agreement.

In 2025, in collaboration with Andrea Esposito, Francesco Rossi, and Marco Bernardo (University of Urbino, Italy) and Francesco Fabris (University of Trieste, Italy), we devised a process algebraic model of the Algorand protocol with the aim of enabling formal verification. Our model captures the behavior of participants in terms of the structured alternation of consensus steps toward a committee-based agreement. We validated the correctness of the protocol in the absence of adversaries and then extended our model to assess the influence of coordinated malicious nodes that can force the commit of an empty block instead of the proposed one. The adversarial scenario was analyzed using CADP through an equivalence-checking-based

noninterference framework, which highlighted both the robustness and the limitations of the Algorand protocol under adversarial assumptions. This work was published as a research report [18].

6.5.5 Mobile Robots

Participants: Radu Mateescu, Wendelin Serwe (*correspondent*), Aline Uwimbabazi.

Robot systems increasingly interact with humans in various domains, such as transport, environment, health, and social care. Indoor navigation of robots poses several challenges. Firstly, the indoor spaces are dynamic: people move, floor might be wet, doors open and close, or the plans for the mobile robot might have been changed. Secondly, the robots should respect constraints such as safety, accessibility, and environment disruptions.

In 2025, in the framework of the A-IQ Ready project (see § 8.2.1), in collaboration with Lina Marsso and Pierre-Yves Lajoie (Polytechnique Montréal, Canada), we considered the formal modeling of the behavior of a mobile robot that interacts with humans in dynamic indoor environments. Specifically, we modeled in LNT the behavior of a robot assisting humans (static and moving) with the food preparation, and calling for help in emergency situations (e.g., when a human has fallen on the floor). The robot behavior is defined by several operations assisting the food preparation (successful move, detection, taking and giving an object) and handling emergency situations (detection, calling for help, giving instructions). Besides the appropriate actions of the robot in each case, the LNT model also comprises the probabilities for the robot's actions to be performed successfully (e.g., the probability of finding the closest human to call for help, or navigating along the shortest path). The LNT model will serve as basis to analyze the safety of robot interactions and to study the variation of action probabilities depending on environmental changes.

7 Bilateral contracts and grants with industry

7.1 Bilateral grants with industry

7.1.1 Euris

Participants: Suraj Gupta, Frédéric Lang, Gwen Salaün (*correspondent*).

S. Gupta is supported by a CIFRE grant (from February 2024 to January 2027) from **Euris** (Paris) on the formal modeling and analysis of blockchain protocols in the health domain, under the supervision of Gwen Salaün, Frédéric Lang, and Umar Ozeer (Euris).

7.1.2 Public IA

Participants: Nabil Bouchta, Gwen Salaün (*correspondent*).

N. Bouchta is supported by a CIFRE grant (from November 2025 to October 2028) from **Public IA** (Illkirch-Graffenstaden) on the formal modeling, analysis, and optimization of business processes, under the supervision of Gwen Salaün, Jean-Michel Bernabotto (Public IA), and Quentin Christoffel (Public IA).

8 Partnerships and cooperations

8.1 International initiatives

H. Garavel is a member of IFIP (*International Federation for Information Processing*) Technical Committee 1 (*Foundations of Computer Science*) Working Group 1.8 on Concurrency Theory chaired successively by

Luca Aceto and Jos Baeten.

8.1.1 Other international collaborations

In 2025, we had scientific relations with several universities and institutes abroad, including:

- University of Oxford, UK (Dave Parker)
- University of Urbino, Italy (Marco Bernardo)
- Saarland University, Germany (Holger Hermanns)
- University of Málaga, Spain (Francisco Durán)
- Polytechnique Montréal, Canada (Lina Marsso)

8.2 European initiatives

8.2.1 Horizon Europe

A-IQ Ready

Participants: Wei Chen, Frédéric Lang, Radu Mateescu (*correspondent*), Wendelin Serwe, Aline Uwimbabazi.

[A-IQ Ready project on cordis.europa.eu](https://cordis.europa.eu)

Title: Artificial Intelligence using Quantum measured Information for realtime distributed systems at the edge

Duration: From January 1, 2023 to December 31, 2025

Partners:

- SCALIRO GMBH, Germany
- BUNDESMINISTERIUM FÜR LANDESVERTEIDIGUNG (FEDERAL MINISTRY OF NATIONAL DEFENSE), Austria
- UAB TERAGLOBUS, Lithuania
- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- SAFELOG GMBH, Germany
- AVL ARASTIRMA VE MUHENDISLIK SANAYI VE TICARET LIMITED SIRKETI (AVL TURKIYE), Türkiye
- EMOTION3D GMBH, Austria
- AIT AUSTRIAN INSTITUTE OF TECHNOLOGY GMBH (AIT), Austria
- SYNOPSIS NETHERLANDS BV (VIRAGE LOGIC), Netherlands
- TECHNISCHE UNIVERSITAET GRAZ (TU GRAZ), Austria
- IDEAS & MOTION SRL, Italy
- CHAROKOPEIO PANEPISTIMIO (HAROKOPIO UNIVERSITY OF ATHENS (HUA)), Greece
- MONTANUNIVERSITAET LEOBEN (Montanuniversitaet Leoben), Austria
- TEKNE SRL (TEKNE), Italy
- SLEEP ADVICE TECHNOLOGIES SRL, Italy
- HUAWEI TECHNOLOGIES SWEDEN AB (HWSE), Sweden

- INFORMATION TECHNOLOGY FOR MARKET LEADERSHIP (ITML), Greece
- UNIKIE OY (UNIKIE), Finland
- MERCEDES-BENZ AG, Germany
- IOBUNDLE, LDA, Portugal
- TEKNOLOGIAN TUTKIMUSKESKUS VTT OY (VTT), Finland
- KOUVOLA INNOVATION OY, Finland
- TECHNISCHE UNIVERSITAET MUENCHEN (TUM), Germany
- INESC TEC - INSTITUTO DE ENGENHARIADE SISTEMAS E COMPUTADORES, TECNOLOGIA E CIENCIA (INESC TEC), Portugal
- INSTITUT MIKROELEKTRONICKYCH APLIKACI SRO (IMA), Czechia
- HOCHSCHULE OFFENBURG, Germany
- TECHNISCHE HOCHSCHULE ROSENHEIM / TECHNICAL UNIVERSITY OF APPLIED SCIENCES (TECHNISCHE HOCHSCHULE ROSENHEIM), Germany
- N-VISION SYSTEMS AND TECHNOLOGIES SL (NVISION), Spain
- VAISTO SOLUTIONS OY, Finland
- INSAR.SK SRO (INSAR.SK), Slovakia
- VYSOKE UCENI TECHNICKE V BRNE (BRNO UNIVERSITY OF TECHNOLOGY), Czechia
- OSTBAYERISCHE TECHNISCHE HOCHSCHULEAMBERG-WEIDEN (OTH Amberg-Weiden), Germany
- MANTSINEN GROUO LTD OY, Finland
- VIRTUAL VEHICLE RESEARCH GMBH (VIF), Austria
- METSA FIBRE OY (POHJAN SELLU KEMI BOTNIA PULPS METSA-RAUMA METSA-BOTNIA), Finland
- INNATERA NANOSYSTEMS BV, Netherlands
- PUMACY TECHNOLOGIES AG (PUMACY), Germany
- KALMAR FINLAND OY (KALMAR), Finland
- UNIVERSIDAD DE ALCALA (UNIVERSIDAD DE ALCALA), Spain
- SILICON MOBILITY (SILICON MOBILITY), France
- POLITECNICO DI TORINO (POLITO), Italy
- AVL LIST GMBH (AVL), Austria
- TTTECH AUTO GMBH, Austria
- TTTECH COMPUTERTECHNIK AG, Austria
- ELEKTRONIKAS UN DATORZINATNU INSTITUTS (EDI), Latvia
- IL-INGENIEURBURO LAABMAYR & PARTNER ZT GESMBH (Laabmayr), Austria
- UNIVERSITAET zu LUEBECK (UZL), Germany
- UNIVERSITA DEGLI STUDI DI MODENA E REGGIO EMILIA (UNIMORE), Italy
- UNIVERSIDAD POLITECNICA DE MADRID (UPM), Spain
- ARQUIMEA RESEARCH CENTER SL, Spain

Inria contact: Radu Mateescu

Coordinator: Katrin Al Jezany (AVL)

Summary: Global environmental issues, social inequality and geopolitical changes will pose numerous problems for our society in the future. To face these new challenges and deal with them, there is a need to understand and appropriately utilize new digital technologies such as artificial intelligence (AI), the Internet of Things (IoT), robotics and biotechnologies.

A-IQ Ready proposes cutting-edge quantum sensing, edge continuum orchestration of AI and distributed collaborative intelligence technologies to implement the vision of intelligent and autonomous ECS for the digital age. Quantum magnetic flux and gyro sensors enable highest sensitivity and accuracy without any need for calibration, offer unmatched properties when used in combination with a magnetic field map. Such a localization system will enhance the timing and accuracy of the autonomous agents and will reduce false alarms or misinformation by means of AI and multi-agent system concepts. As a priority, the communication guidance and decision making of groups of agents need to be based on cutting-edge technologies. Edge continuum orchestration of AI will allow decentralizing the development of applications, while ensuring an optimal use of the available resources. Combined with the quantum sensors, the edge continuum will be equipped with innovative, multi-physical capabilities to sense the environment, generating “slim” but accurate measurements. Distributed intelligence will enable emergent behavior and massive collaboration of multiple agents towards a common goal. By exploring the synergies of these cutting-edge technologies through civil safety and security, digital health, smart logistics for supply chains and propulsion use cases, A-IQ Ready will provide the basis for the digital society in Europe based on values, moving towards the ideal of Society 5.0.

The main contributions of CONVECS are the formal modeling and validation of intelligent transportation systems and indoor logistics applications.

8.2.2 Other european programs/initiatives

The CONVECS project-team is member of the **FMICS** (*Formal Methods for Industrial Critical Systems*) working group of **ERCIM**. H. Garavel and R. Mateescu are members of the FMICS board, H. Garavel being in charge of dissemination actions.

8.3 National initiatives

8.3.1 PEPR Cloud

The cloud has become an essential ingredient of IT systems. Its model, initially based on large data centers, has evolved towards “edge computing” or “digital continuum”, developing secure, interoperable hybrid clouds to process large volumes of data quickly and efficiently. This change is crucial for future applications such as smart cities or autonomous vehicles. However, cloud software needs to be rethought to adapt to the diversity of operators, multiple access points, user and resource mobility, while addressing the challenges of security and energy consumption control.

The **PEPR Cloud** is a research program promoted by the French government as part of the France 2030 “Cloud” strategy to develop secure and frugal Cloud technologies. PEPR Cloud aims to advance Cloud technologies and facilitate the transfer of innovations and solutions from research to industry. PEPR Cloud started in April 2024 for seven years. CONVECS is involved in two projects of PEPR Cloud, described below.

Archi-CESAM

Participants: Zachary Assoumani, Hubert Garavel, Radu Mateescu (*correspondent*), Wendelin Serwe.

Archi-CESAM (*Converged, Efficient and Safe Architecture based on Near Memory Accelerators*) is a PEPR Cloud project led by CEA and involving five other partners (Inria, Université de Rennes, Télécom SudParis, Grenoble INP, and CNRS). The project proposes to rethink hardware (computation, memory and interconnection) so that it is co-designed with the application in a converged and trusted architecture perspective, in an environment known for its abundance of data to be processed. Archi-CESAM tackles the

major cloud evolution (increased parallelism, specialization, new interconnections, virtualization) with a global, coordinated approach to distributed architectures, acceleration, interconnection and security bricks, not forgetting design methods.

The main contribution of CONVECS to Archi-CESAM is a rigorous methodology of designing hardware architectures, based on formal methods, verification, and conformance test generation.

TARANIS

Participants: Ahmed Khebbab, Gwen Salaün (*correspondent*).

TARANIS (*Model, Deploy, Orchestrate, and Optimize Cloud Applications and Infrastructure*) is a PEPR Cloud project led by Inria Lyon and involving nine other partners (CNRS, IMT, UGA, CEA, Université de Rennes, ENS Lyon, Université Claude Bernard, Université de Lille, INSA Rennes). The project proposes to exploit the new cloud infrastructures (edge computing, digital continuum) efficiently by abstracting the description of applications and resources to automate their management even further. This will enable a global optimization of resources according to multi-criteria objectives (price, deadline, performance, energy, etc.) on both the user side (applications) and the resource provider side (infrastructures). TARANIS also addresses the challenges of abstracting application reconfiguration and dynamically adapting resource usage.

The main contribution of CONVECS to TARANIS is on resource provisioning and orchestration languages and platforms that are used for deploying and updating cloud applications. More precisely, we plan to develop formal specifications and models as well as analysis capabilities for verifying functional and non-functional properties on such models.

8.3.2 Other national collaborations

We had sustained scientific relations with the following researchers:

- César Fuguet (MADMAX project-team)

8.4 Regional initiatives

8.4.1 Région Auvergne-Rhône-Alpes

Participants: Gwen Salaün (*correspondent*).

MOAP is a project funded by the Auvergne-Rhône-Alpes region within the *Pack Ambition Recherche* programme. The project involves the project-teams CONVECS and CORSE, and the SOITEC company. MOAP aims at providing modelling and automated analysis techniques for enabling companies to master the complexity of their internal processes and for optimizing those processes with the final goal of improving the quality and productivity of their businesses.

MOAP started in October 2020 for five years. The main contributions of CONVECS to MOAP are the formal modeling and automated verification of BPMN processes.

9 Dissemination

Participants: Zachary Assoumani, Hubert Garavel, Ahmed Khebbab, Frédéric Lang, Radu Mateescu (*correspondent*), Quentin Nivon, Gwen Salaün, Wendelin Serwe, Aline Uwimbabazi.

9.1 Promoting scientific activities

9.1.1 Scientific events: organisation

General chair, scientific chair

- Together with Peter Höfner (Data61, CSIRO, Sydney, Australia), H. Garavel set up a model repository to collect and archive formal models of real systems; this infrastructure is used by the series of **MARS** workshops. This repository currently contains 37 models, among which 11 were deposited by CONVECS.
- H. Garavel is a member of the model board of **MCC** (*Model Checking Contest*).
- H. Garavel is a member of the steering committee of the MARS (*Models for Formal Analysis of Real Systems*) workshop series since 2015.
- H. Garavel is a member of the steering committee of TTC (*Transformation Tool Contest*) since 2021.
- H. Garavel and R. Mateescu are members of the steering committee of the FMICS (*Formal Methods for Industrial Critical Systems*) conference series since 2018.
- G. Salaün is member of the steering committee of the FACS (*Formal Aspects of Component Software Symposium*) conference series since 2021.
- G. Salaün is member of the steering committee of the ACM SAC-SVT (*Symposium of Applied Computing – Software Verification and Testing track*) conference series since 2018.
- G. Salaün is member of the steering committee of the SEFM (*International Conference on Software Engineering and Formal Methods*) conference series since 2014.

Member of the organizing committees

- A. Khebbeb was publicity chair of DATAMOD'2025 (*13th International Symposium “From Data to Models and Back”*), Toledo, Spain, November 10-11, 2025.
- Q. Nivon was Web chair of FormaliSE'2025 (*13th International Conference on Formal Methods in Software Engineering*), Ottawa, Ontario, Canada, April 27-28, 2025.

9.1.2 Scientific events: selection

Chair of conference program committees

- G. Salaün was programme committee co-chair of FormaliSE'2025.
- G. Salaün was programme committee co-chair of DATAMOD'2025.

Member of the conference program committees

- H. Garavel was a programme committee member of FMICS'2025 (*30th International Conference on Formal Methods for Industrial Critical Systems*), Aarhus, Denmark, August 25-30, 2025.
- A. Khebbeb was an artifact evaluation committee member of ICSOC'2025 (*23rd International Conference on Service-Oriented Computing*), Shenzhen, China, December 1-4, 2025.
- F. Lang was a programme committee member of SEFM'2025.
- R. Mateescu was a programme committee member of ASQAP'2025 (*1st International Workshop on Autonomous System Quality Assurance and Prediction with Digital Twins*), Hamilton, Canada, May 4, 2025.
- R. Mateescu was a programme committee member of SPIN'2025 (*31st International Symposium on Model Checking Software*), Hamilton, Canada, May 7-8, 2025.

- R. Mateescu was a programme committee member of ICTSS'2025 (*37th International Conference on Testing Software and Systems*), Limassol, Cyprus, September 17-19, 2025.
- G. Salaün was a programme committee member of WAKA'2025 (*Workshop on Adaptable Cloud Architectures*), Lille, France, June 20, 2025.
- G. Salaün was a programme committee member of SAC-SVT'2025 (*40th ACM/SIGAPP Symposium on Applied Computing - Software Verification and Testing Track*), Catania, Sicily, March 31 - April 4, 2025.
- G. Salaün was a programme committee member of SEFM'2025.
- G. Salaün was a programme committee member of FormaliSE'2025.
- G. Salaün was a programme committee member of DATAMOD'2025.
- W. Serwe was a programme committee member of FMICS'2025.
- W. Serwe was a programme committee member of ASYDE'2025 (*7th International Workshop on Automated and verifiable Software sYstem DEvelopment*), Seoul, South Korea, November 16, 2025.
- A. Uwimbabazi was an artifact evaluation committee member of SPIN'2025.

Reviewer

- Z. Assoumani was a reviewer for FMICS'2025.
- A. Khebbeb was a reviewer for DATAMOD'2025 and SEFM'2025.
- Q. Nivon was a reviewer for DATAMOD'2025, SAC-SVT'2025, and SEFM'2025.
- W. Serwe was a reviewer for SPIN'2025.

9.1.3 Journal

Member of the editorial boards

- H. Garavel is an editorial board member of STTT (*Springer International Journal on Software Tools for Technology Transfer*).

Reviewer - reviewing activities

- F. Lang was a reviewer for PeerJ Computer Science and STTT.
- R. Mateescu was a reviewer for SCP (*Science of Computer Programming*) and TR (*IEEE Transactions on Reliability*).
- W. Serwe was a reviewer for STTT, JSS (*Journal of Systems and Software*), and EAAI (*Engineering Applications of Artificial Intelligence*).

9.1.4 Software dissemination and internet visibility

The CONVECS project-team distributes several software tools, among which the CADP toolbox. In 2025, the main facts are the following:

- We prepared and distributed twelve successive versions (2025-a to 2025-l) of CADP.
- We granted CADP licenses for 171 different computers in the world.

The CONVECS Web site was updated with scientific contents, announcements, publications, etc.

By the end of December 2025, the CADP forum, opened in 2007 for discussions regarding the CADP toolbox, had over 483 registered users and over 2014 messages had been exchanged.

Also, for the 2025 edition of the Model Checking Contest, we provided 3 families of models (totalling 54 Nested-Unit Petri Nets) derived from our LNT models.

Other teams also used the CADP toolbox for various case studies:

- Checking model consistency in service-oriented systems [33]
- Timing analysis of service-oriented architectures in software-defined vehicles [40]
- Bridging threat models and detections using formal verification [39]
- Identifying vulnerabilities of operational technology protocols in industrial IoT [20]

9.1.5 Invited talks

- Z. Assoumani gave a talk entitled “*Connecting Hardware Description Languages and Formal Languages*” at the Archi-CESAM project workshop held at Inria Paris on November 5, 2025.
- H. Garavel gave a talk entitled “*Formal Study of Algorand’s Byzantine Agreement Algorithm*” at the seminar of the MTV2 (*Méthodes de test pour la validation et la vérification*) working group of the GDR GPL, held at Grenoble, France, on December 11, 2025.
- R. Mateescu gave a talk entitled “*Improving PSS Test Generation Using Model Checking and Conformance Testing*” at the scientific seminar of the PEPR Cloud held online on September 19, 2025.
- Q. Nivon gave a talk entitled “*LLM-Based Generation of BPMN Workflows From Textual Descriptions*” at the “BPM & IA” workshop organized by G. Salaün at the LIG laboratory on May 14, 2025.
- G. Salaün gave a talk entitled “*Modelling, Runtime Analysis and Optimization of Business Processes*” at the seminar of the MTV2 (*Méthodes de test pour la validation et la vérification*) working group of the GDR GPL, held at Grenoble, France, on December 11, 2025.
- G. Salaün gave a talk entitled “*Quantitative Analysis and Runtime Enforcement for IEC 61499*” at the 11th International Summer School on Industrial Agents (ISSIA’25) held at Ancona, Italie, on June 30 – July 4, 2025.
- G. Salaün gave a talk entitled “*Modelling, Analysis and Optimization of BPMN Processes*” at IRIT, Toulouse, France, on June 10, 2025.
- G. Salaün gave a talk entitled “*LLM-Based Generation of BPMN Workflows From Textual Descriptions*” at ANITI, Toulouse, France, on June 10, 2025.

9.1.6 Research administration

- R. Mateescu is the scientific correspondent of the International Partnerships for Inria Grenoble.
- R. Mateescu was a member of the “*Comité d’Orientation Scientifique*” for Inria Grenoble until August 31, 2025.
- R. Mateescu is representative of Inria Grenoble at the International Relations and Outreach of Université Grenoble Alpes (UGA).
- G. Salaün is the director of the MSTIC research department of UGA.
- G. Salaün was the president of the scientific evaluation committee “*Sciences et génie du logiciel, Réseaux de communication multi-usages et infrastructures de haute performance*” of ANR (CE25) from 2022 to 2025.

- W. Serwe is the chair of the “*Commission du développement technologique*”, which is in charge of selecting R&D projects for Inria Grenoble, and giving an advice on the recruitment of temporary engineers.
- W. Serwe is a member of the “*Comité de Centre*” at Inria Grenoble.

9.2 Teaching - Supervision - Juries - Educational and pedagogical outreach

9.2.1 Teaching

CONVECS is a host team for the computer science master MOSIG (*Master of Science in Informatics at Grenoble*), common to Grenoble INP and UGA.

In 2025, we carried out the following teaching activities:

- Z. Assoumani gave a course on “*Théorie des langages 2*” (18 hours “*équivalent TD*”) to first year students of ENSIMAG.
- Z. Assoumani gave a course on “*Algorithmique distribuée*” (15 hours “*équivalent TD*”) to fourth year students of Polytech.
- Z. Assoumani gave a course on “*Sémantique des langages de programmation & compilation*” (30 hours “*équivalent TD*”) to M1 students of UGA.
- Z. Assoumani gave a course on “*Automates et langages*” (29 hours “*équivalent TD*”) to L2 students of UGA.
- F. Lang gave a course on “*Formal Software Development Methods*” (7.5 hours “*équivalent TD*”) in the framework of the “*Software Engineering*” lecture given to first year students of the MOSIG.
- F. Lang gave a course on “*Programming Languages, Compilers, and Semantics*” (27 hours “*équivalent TD*”) to first year students of the MOSIG and of the Master 1 Informatique.
- F. Lang and R. Mateescu gave a lecture on “*Modeling and Analysis of Concurrent Systems: Models and Languages for Model Checking*” (27 hours “*équivalent TD*”) to third year students of ENSIMAG.
- F. Lang gave a course on “*Modeling and Analysis of Asynchronous Concurrent Systems*” (15 hours “*équivalent TD*”) in the framework of the “*Safety Critical Systems*” lecture given to second year students of the MOSIG.
- G. Salaün taught about 100 hours of classes (algorithmics, Web development, object-oriented programming) at the MMI department of IUT1/UGA.
- W. Serwe supervised a group of six teams in the context of the “*projet Génie Logiciel*” (55 hours “*équivalent TD*”, consisting in 13.5 hours of lectures, plus supervision and evaluation), ENSIMAG, January 2025.

9.2.2 Supervision

- PhD: Q. Nivon, “*Analysis, Optimisation and Debugging of BPMN Processes*”, Université Grenoble Alpes, defended on December 12, 2025, G. Salaün
- PhD in progress: Z. Assoumani, “*Conception rigoureuse de circuits basée sur les méthodes formelles*”, Université Grenoble Alpes, since October 2024, R. Mateescu and W. Serwe
- PhD in progress: N. Bouchta, “*Modélisation formelle, analyse et optimisation de processus métier*”, Université Grenoble Alpes, since November 2025, G. Salaün, Jean-Michel Bernabotto (Public IA), and Quentin Christoffel (Public IA)
- PhD in progress: S. Gupta, “*Using blockchains for managing EHRs (Electronic Health Records)*”, Université Grenoble Alpes, since January 2024, G. Salaün, F. Lang, and Umar Ozeer (Euris)

- PhD in progress: J-B. Horel, “*Validation des composants de perception basés sur l’IA dans les véhicules autonomes*”, Université Grenoble Alpes, since April 2021, R. Mateescu, Alessandro Renzaglia, and Christian Laugier (CHROMA project-team)
- PhD in progress: A. Khebbeb, “*Formal Modelling and Automated Analysis of Resource Provisioning Languages*”, Université Grenoble Alpes, since December 2024, G. Salaün and Philippe Merle (SPIRALS project-team, Lille)

9.2.3 Juries

- R. Mateescu was reviewer of Dumitru-Bogdan Prelipcean’s PhD thesis, entitled “*Applications des méthodes formelles à la détection de logiciels malveillants*”, defended at Université Paris-Est Créteil on September 12, 2025.

10 Scientific production

10.1 Major publications

- [1] G. Barbon, V. Leroy and G. Salaün. ‘Debugging of Behavioural Models using Counterexample Analysis’. In: *IEEE Transactions on Software Engineering* 47.6 (June 2021), pp. 1184–1197. DOI: [10.1109/TSE.2019.2915303](https://doi.org/10.1109/TSE.2019.2915303). URL: <https://inria.hal.science/hal-02145610>.
- [2] X. Etchevers, G. Salaün, F. Boyer, T. Coupaye and N. De Palma. ‘Reliable Self-deployment of Distributed Cloud Applications’. In: *Software: Practice and Experience* 47.1 (2017), pp. 3–20. DOI: [10.1002/spe.2400](https://doi.org/10.1002/spe.2400). URL: <https://hal.inria.fr/hal-01290465>.
- [3] H. Evrard and F. Lang. ‘Automatic Distributed Code Generation from Formal Models of Asynchronous Processes Interacting by Multiway Rendezvous’. In: *Journal of Logical and Algebraic Methods in Programming* 88 (Mar. 2017), p. 33. DOI: [10.1016/j.jlamp.2016.09.002](https://doi.org/10.1016/j.jlamp.2016.09.002). URL: <https://hal.inria.fr/hal-01412911>.
- [4] H. Gavel. ‘Nested-unit Petri nets’. In: *Journal of Logical and Algebraic Methods in Programming* 104 (Apr. 2019), pp. 60–85. DOI: [10.1016/j.jlamp.2018.11.005](https://doi.org/10.1016/j.jlamp.2018.11.005). URL: <https://hal.inria.fr/hal-02072190>.
- [5] H. Gavel, F. Lang and R. Mateescu. ‘Compositional Verification of Asynchronous Concurrent Systems using CADP’. In: *Acta Informatica* 52.4 (June 2015), p. 56. DOI: [10.1007/s00236-015-0226-1](https://doi.org/10.1007/s00236-015-0226-1). URL: <https://hal.inria.fr/hal-01247507>.
- [6] H. Gavel, F. Lang, R. Mateescu and W. Serwe. ‘CADP 2011: A Toolbox for the Construction and Analysis of Distributed Processes’. In: *International Journal on Software Tools for Technology Transfer* 15.2 (2013), pp. 89–107. DOI: [10.1007/s10009-012-0244-z](https://doi.org/10.1007/s10009-012-0244-z). URL: <http://hal.inria.fr/hal-00715056> (cit. on p. 10).
- [7] H. Gavel, F. Lang and W. Serwe. ‘From LOTOS to LNT’. In: *ModelEd, TestEd, TrustEd - Essays Dedicated to Ed Brinksma on the Occasion of His 60th Birthday*. Ed. by J.-P. Katoen, R. Langerak and A. Rensink. Vol. 10500. Lecture Notes in Computer Science. Springer, Oct. 2017, pp. 3–26. DOI: [10.1007/978-3-319-68270-9_1](https://doi.org/10.1007/978-3-319-68270-9_1). URL: <https://hal.inria.fr/hal-01621670> (cit. on pp. 6, 12).
- [8] A. Krishna, P. Poizat and G. Salaün. ‘Checking Business Process Evolution’. In: *Science of Computer Programming* 170 (Jan. 2019), pp. 1–26. DOI: [10.1016/j.scico.2018.09.007](https://doi.org/10.1016/j.scico.2018.09.007). URL: <https://hal.inria.fr/hal-01920273>.
- [9] R. Mateescu and W. Serwe. ‘Model Checking and Performance Evaluation with CADP Illustrated on Shared-Memory Mutual Exclusion Protocols’. In: *Science of Computer Programming* (Feb. 2012). DOI: [10.1016/j.scico.2012.01.003](https://doi.org/10.1016/j.scico.2012.01.003). URL: <http://hal.inria.fr/hal-00671321>.

10.2 Publications of the year

International peer-reviewed conferences

- [10] A. Cuci, U. Ozeer and G. Salaün. ‘Modelling and Verification of an Application for Managing Sensitive Health Data’. In: *DataMod 2023 - 11th International Symposium on From Data to Models and Back*. Vol. 14618. Lecture Notes in Computer Science. Eindhoven, Netherlands: Springer Nature Switzerland, 15th Apr. 2025, pp. 127–141. DOI: [10.1007/978-3-031-87217-4_7](https://doi.org/10.1007/978-3-031-87217-4_7). URL: <https://hal.science/hal-05066343>.
- [11] S. Gupta, F. Lang, U. Ozeer and G. Salaün. ‘Blockchain as a Service for Electronic Health Records’. In: *Proceedings of the IEEE International Conference on Digital Health (ICDH 2025)*. ICDH 2025 - IEEE International Conference on Digital Health. Helsinki, Finland: IEEE, 2025, pp. 91–101. DOI: [10.1109/ICDH67620.2025.00022](https://doi.org/10.1109/ICDH67620.2025.00022). URL: <https://inria.hal.science/hal-05393417> (cit. on p. 16).
- [12] J.-B. Horel, P. Ledent, R. Mateescu, W. Serwe and A. Uwimbabazi. ‘Assessing Test Scenarios for Autonomous Driving Using Probabilistic Model Checking’. In: *Lecture Notes in Computer Science. ICTSS 2025 - 37th IFIP WG 6.1 International Conference on Testing Software and Systems*. Vol. 16107. Lecture Notes in Computer Science. Limassol, Cyprus: Springer Nature Switzerland, 16th Sept. 2026, pp. 273–289. DOI: [10.1007/978-3-032-05188-2_18](https://doi.org/10.1007/978-3-032-05188-2_18). URL: <https://inria.hal.science/hal-05267555> (cit. on p. 18).
- [13] D. Kaufmann, R. Mateescu, L. Muller, W. Serwe and F. Wotawa. ‘Formal Methods for Residual Risk Reduction in Cyber-Physical Systems’. In: *QRS 2025 - 25th International Conference on Software Quality, Reliability and Security*. Hangzhou, China: IEEE, 2025, pp. 258–269. DOI: [10.1109/QRS65678.2025.00035](https://doi.org/10.1109/QRS65678.2025.00035). URL: <https://inria.hal.science/hal-05305293>.
- [14] Q. Nivon, G. Salaün and F. Lang. ‘GIVUP: Automated Generation and Verification of Textual Process Descriptions’. In: *FSE 2025 - 33rd ACM International Conference on the Foundations of Software Engineering*. Trondheim, Norway, 2025, pp. 1–5. DOI: [10.1145/3696630.3728593](https://doi.org/10.1145/3696630.3728593). URL: <https://inria.hal.science/hal-05131967> (cit. on p. 13).

Conferences without proceedings

- [15] D. Cremer, B. Dalmas, Q. Nivon and G. Salaün. ‘Incremental Synchronization of BPMN Models and Documentations by Leveraging Structural Algorithms and LLMs’. In: *CoopIS 2025 - International Conference on Cooperative Information Systems*. Marbella, Spain, 2025, pp. 1–18. URL: <https://inria.hal.science/hal-05328360> (cit. on p. 14).

Scientific book chapters

- [16] H. Garavel and H. Hermanns. ‘Nondeterminism in Interactive Markov Chains, with Application to the Erlangen Mainframe’. In: *Principles of Formal Quantitative Analysis*. Vol. 15760. Lecture Notes in Computer Science. Springer Nature Switzerland, 30th Aug. 2026, pp. 15–69. DOI: [10.1007/978-3-031-97439-7_2](https://doi.org/10.1007/978-3-031-97439-7_2). URL: <https://inria.hal.science/hal-05265477> (cit. on p. 17).

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

- [17] *Formal methods for industrial critical systems 27.5* (14th Oct. 2025). DOI: [10.1007/s10009-025-00830-0](https://doi.org/10.1007/s10009-025-00830-0). URL: <https://inria.hal.science/hal-05391364>.

Reports & preprints

- [18] A. Esposito, F. Rossi, M. Bernardo, F. Fabris and H. Garavel. *Formal Modeling and Verification of the Algorand Consensus Protocol in CADP*. ArXiv, 2025, pp. 1–25. DOI: [10.48550/arXiv.2508.19452](https://doi.org/10.48550/arXiv.2508.19452). URL: <https://inria.hal.science/hal-05265521> (cit. on p. 20).

Other scientific publications

- [19] Z. Assoumani, C. Fuguet, R. Mateescu and W. Serwe. ‘On Benefits of Modeling the HPDcache in LNT’. In: RISC-V 2025 -RISC-V Summit Europe. Paris, France, 2025, pp. 1–1. URL: <https://inria.hal.science/hal-05106202> (cit. on p. 19).

10.3 Cited publications

- [20] M. Boeding, M. Hempel and H. Sharif. ‘End-to-End Framework for Identifying Vulnerabilities of Operational Technology Protocols and Their Implementations in Industrial IoT’. In: *Future Internet* 17.34 (2025). DOI: [10.3390/fi17010034](https://doi.org/10.3390/fi17010034) (cit. on p. 27).
- [21] D. Champelovier, X. Clerc, H. Garavel, Y. Guerte, C. McKinty, V. Powazny, F. Lang, W. Serwe and G. Smeding. ‘Reference Manual of the LNT to LOTOS Translator (Version 6.8)’. INRIA, Grenoble, France. Jan. 2019 (cit. on p. 12).
- [22] E. M. Clarke, E. A. Emerson and A. P. Sistla. ‘Automatic Verification of Finite-State Concurrent Systems using Temporal Logic Specifications’. In: *ACM Transactions on Programming Languages and Systems* 8.2 (Apr. 1986), pp. 244–263 (cit. on p. 11).
- [23] R. De Nicola and F. W. Vaandrager. ‘Action versus State Based Logics for Transition Systems’. In: *Semantics of Concurrency*. Vol. 469. Lecture Notes in Computer Science. Springer Verlag, 1990, pp. 407–419 (cit. on p. 11).
- [24] H. Garavel. ‘Compilation of LOTOS Abstract Data Types’. In: *Proceedings of the 2nd International Conference on Formal Description Techniques FORTE’89 (Vancouver B.C., Canada)*. Ed. by S. T. Vuong. North Holland, Dec. 1989, pp. 147–162 (cit. on p. 10).
- [25] H. Garavel. ‘OPEN/CÆSAR: An Open Software Architecture for Verification, Simulation, and Testing’. In: *Proceedings of the First International Conference on Tools and Algorithms for the Construction and Analysis of Systems TACAS’98 (Lisbon, Portugal)*. Ed. by B. Steffen. Vol. 1384. Lecture Notes in Computer Science. Full version available as INRIA Research Report RR-3352. Berlin: Springer Verlag, Mar. 1998, pp. 68–84 (cit. on p. 8, 10).
- [26] H. Garavel and F. Lang. ‘SVL: a Scripting Language for Compositional Verification’. In: *Proceedings of the 21st IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems FORTE’2001 (Cheju Island, Korea)*. Ed. by M. Kim, B. Chin, S. Kang and D. Lee. Full version available as INRIA Research Report RR-4223. IFIP. Kluwer Academic Publishers, Aug. 2001, pp. 377–392 (cit. on p. 11).
- [27] H. Garavel, F. Lang and R. Mateescu. ‘Compiler Construction using LOTOS NT’. In: *Proceedings of the 11th International Conference on Compiler Construction CC 2002 (Grenoble, France)*. Ed. by N. Horspool. Vol. 2304. Lecture Notes in Computer Science. Springer Verlag, Apr. 2002, pp. 9–13 (cit. on p. 12).
- [28] H. Garavel, R. Mateescu and I. Smarandache-Sturm. ‘Parallel State Space Construction for Model-Checking’. In: *Proceedings of the 8th International SPIN Workshop on Model Checking of Software SPIN’2001 (Toronto, Canada)*. Ed. by M. B. Dwyer. Vol. 2057. Lecture Notes in Computer Science. Revised version available as INRIA Research Report RR-4341 (December 2001). Berlin: Springer Verlag, May 2001, pp. 217–234 (cit. on p. 11).
- [29] H. Garavel and W. Serwe. ‘State Space Reduction for Process Algebra Specifications’. In: *Theoretical Computer Science* 351.2 (Feb. 2006), pp. 131–145 (cit. on p. 10).
- [30] H. Garavel and J. Sifakis. ‘Compilation and Verification of LOTOS Specifications’. In: *Proceedings of the 10th International Symposium on Protocol Specification, Testing and Verification (Ottawa, Canada)*. Ed. by L. Logrippo, R. L. Probert and H. Ural. IFIP. North Holland, June 1990, pp. 379–394 (cit. on p. 10).
- [31] M. Hennessy and R. Milner. ‘Algebraic Laws for Nondeterminism and Concurrency’. In: *Journal of the ACM* 32 (1985), pp. 137–161 (cit. on p. 11).
- [32] H. Hermanns. *Interactive Markov Chains and the Quest for Quantified Quality*. Vol. 2428. Lecture Notes in Computer Science. Springer Verlag, 2002 (cit. on p. 17).

- [33] H. Jiang and K. Kontogiannis. ‘Checking Model Consistency in Service-Oriented Systems’. In: *Proceedings of the 12th Annual IEEE International systems Conference (SysCon’2025), Montréal, QC, Canada*. IEEE CS Press, 2025, pp. 1–8. doi: [10.1109/SysCon64521.2025.11014829](https://doi.org/10.1109/SysCon64521.2025.11014829) (cit. on p. 27).
- [34] J. Magee and J. Kramer. *Concurrency: State Models and Java Programs*. 2006th ed. Wiley, Apr. 2006 (cit. on p. 11).
- [35] L. Marsso, R. Mateescu and W. Serwe. ‘TESTOR: A Modular Tool for On-the-Fly Conformance Test Case Generation’. In: *TACAS 2018 - 24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Vol. 10806. Lecture Notes in Computer Science. Thessaloniki, Greece: Springer, Apr. 2018, pp. 211–228. doi: [10.1007/978-3-319-89963-3_13](https://doi.org/10.1007/978-3-319-89963-3_13). URL: <https://hal.inria.fr/hal-01777861> (cit. on p. 18).
- [36] R. Mateescu and D. Thivolle. ‘A Model Checking Language for Concurrent Value-Passing Systems’. In: *Proceedings of the 15th International Symposium on Formal Methods FM’08 (Turku, Finland)*. Ed. by J. Cuellar, T. Maibaum and K. Sere. Vol. 5014. Lecture Notes in Computer Science. Springer Verlag, May 2008, pp. 148–164 (cit. on p. 7).
- [37] Q. Nivon. ‘Analysis, Optimisation and Debugging of BPMN Processes’. PhD Thesis. Université Grenoble Alpes, Dec. 2025 (cit. on p. 13).
- [38] M. Prédhumeau. ‘Modélisation et simulation de comportements piétons réalistes en espace partagé avec un véhicule autonome’. Theses. Université Grenoble Alpes [2020-....], Dec. 2021. URL: <https://hal.science/tel-03518751> (cit. on p. 18).
- [39] D.-B. Prelicean and C. Dima. ‘Bridging Threat Models and Detections: Formal Verification via CADP’. In: *Proceedings of the 9th edition of the Working Formal Methods Symposium (FROM’2025), Iași, Romania*. Vol. 427. EPTCS. Sept. 2025, pp. 59–78 (cit. on p. 27).
- [40] P. Tokariev, I. Faqrizal and J. Deantoni. ‘Understandable Timing Analysis of Service-Oriented Architecture Components in Software-Defined Vehicle’. In: *Communications in Computer and Information Science. CCIS. Proceedings of the 20th Int. Conf. on Information and Communication Technologies in Education, Research, and Industrial Applications (ICTERI-2025) CCIS-2359*. Nice, France, Sept. 2025. URL: <https://inria.hal.science/hal-05224373> (cit. on p. 27).
- [41] C. F. Tortolero. ‘HPDcache: Open-Source High-Performance L1 Data Cache for RISC-V Cores’. In: *Proceedings of the 20th ACM International Conference on Computing Frontiers (CF’2023), Bologna, Italy*. Ed. by A. Bartolini, K. F. D. Rietveld, C. D. Schuman and J. Moreira. ACM, May 2023, pp. 377–378. doi: [10.1145/3587135.3591413](https://doi.org/10.1145/3587135.3591413) (cit. on p. 19).