RESEARCH CENTRE

Inria Centre
at Université Grenoble Alpes

IN PARTNERSHIP WITH:

Université de Grenoble Alpes

2023
ACTIVITY REPORT

Project-Team

CONVECS

Construction of verified concurrent systems

IN COLLABORATION WITH: Laboratoire d'Informatique de Grenoble (LIG)

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Proofs and Verification

Contents

Project-Team CONVECS

Creation of the Project-Team: 2014 January 01

Keywords

Computer sciences and digital sciences

A1.3.5. – Cloud

A2.1.1. – Semantics of programming languages

A2.1.6. – Concurrent programming

A2.1.7. – Distributed programming

A2.4.1. – Analysis

A2.4.2. – Model-checking

A2.5. – Software engineering

A2.5.1. – Software Architecture & Design

A2.5.4. – Software Maintenance & Evolution

A2.5.5. – Software testing

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

Other research topics and application domains

B5.1. – Factory of the future

B5.4. – Microelectronics

B6.1.1. – Software engineering

B6.3.2. – Network protocols

B6.4. – Internet of things

B6.5. – Information systems

B6.6. – Embedded systems

B7.2.1. – Smart vehicles

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]

Faculty Member

- Gwen Salaün [Univ Grenoble Alpes, Professor, HDR]

Post-Doctoral Fellow

- Aline Uwimbabazi [Schlumberger Foundation, from Aug 2023]

PhD Students

- Pierre Bouvier [Inria, from Feb 2023 until Apr 2023]
- Pierre Bouvier [Univ Grenoble Alpes, until Jan 2023]
- Irman Faqrizal [Univ Grenoble Alpes]
- Philippe Ledent [Inria, from Oct 2023]
- Philippe Ledent [STMicroelectronics, until Sep 2023]
- Lucie Muller [Inria]
- Quentin Nivon [Univ Grenoble Alpes]
- Ahang Zuo [Univ Grenoble Alpes, ATER, from Oct 2023]
- Ahang Zuo [Univ Grenoble Alpes, until Sep 2023]

Interns and Apprentices

- Suraj Gupta [Inria, Intern, from Feb 2023 until Jul 2023]
- Carlos Uriel Vargas Lopez [Inria, Intern, until Apr 2023]

Administrative Assistant

- Myriam Etienne [Inria]

External Collaborators

- Pierre Boullier [retired Inria senior researcher]
- Francisco Duran [University of Malaga, until Mar 2023]
- Dasarada Ramu Munnangi [Inria, until Jun 2023]
- Ajay Krishna Muroor-Nadumane [Inria]

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 [6] 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*) [35], 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 ergonomy 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 [26], 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 § 7.5) 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, avionic 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, electronic transactions, cryptographic key distribution,
- *Telecommunications:* high-speed networks, network management, mobile telephony, feature interaction detection.

5 Highlights of the year

5.1 Awards

H. Garavel, R. Mateescu, F. Lang, and W. Serwe received the ETAPS Test-of-Time Tool Award in April 2023 for their work on developing, enhancing, and maintaining the CADP toolbox. This award was established by ETAPS to acknowledge the importance of reliable and well-maintained research tools having a lasting effect on the community, and the significant effort that their creation and maintenance entails.

5.2 Contests

The Voyance Systems start-up project won the i-PhD 2023 innovation contest organized by BPI France within the France 2030 programme.

6 New software, platforms, open data

6.1 New software

6.1.1 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*) [5] is a toolbox for protocols and distributed systems engineering.

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

- CAESAR.ADT [25] 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 [31, 30] 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 [26] 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 [32], CTL[20], ACTL[21], etc.) to be defined in XTL.

- PBG (*Partitioned BCG Graph*) is a file format implementing the theoretical concept of *Partitioned LTS* [29] 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 [33] 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 [27] 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

6.1.2 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 [28], 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

7 New results

7.1 New Formal Languages and their Implementations

7.1.1 LNT and LOTOS NT Specification Languages

Participants: Hubert Garavel, Frédéric Lang, Wendelin Serwe.

LNT [6] [18] 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 § 7.5). 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. Since TRAIAN 3.0, the lexer and parser of TRAIAN are built using the SYNTAX compiler-generation system developed at Inria Paris and the abstract syntax tree of LOTOS NT, the library of predefined LOTOS NT types and functions, the static semantics checking (identifier binding, type checking, dataflow analysis, etc.), and the C code generation are implemented in LOTOS NT itself, so that TRAIAN is capable of bootstrapping itself.

In 2023, our efforts led to the release of four new major versions of TRAIAN, which introduce various changes to the LOTOS NT language in order to further align it with LNT:

- TRAIAN 3.9 brings twelve language changes, mostly improving the readability of value expressions and patterns, and extending the predefined libraries with new operators, as well as ten static-semantics changes refining the dataflow analysis and extending the class of accepted constructs (record field manipulation, patterns, set notations). Four bugs were fixed.

- TRAIAN 3.10 brings seven language changes enhancing pragmas, “with” clauses, and patterns, as well as six library changes improving the implementation of basic types. Thirteen bugs were fixed.

- TRAIAN 3.11 brings three language changes improving value expressions and the “of” notation, as well as fourteen library changes enhancing basic, array, and set types. Seven bugs were fixed. This release achieves a complete convergence between LOTOS NT and LNT. Measured on a test suite of 13,600+ correct LNT programs totalling more than 9 million non-blank lines of LNT code, TRAIAN 3.11 accepts 100% of them, both syntactically and semantically.

- TRAIAN 3.12 brings eight static-semantics changes refining synchronized events, parallel composition, "case" instructions, and record field manipulation, as well as seven library changes improving the list and set types. Twelve bugs were fixed.

Each version of TRAIAN includes various compiler and code-generation changes accomodating the newly introduced features. For each version, the “traian_upc” conversion tool was extended to ease the upgrade of existing LOTOS NT source code whenever possible, and the user manual of TRAIAN, as well as the demo examples, were constantly updated.

The LNT language has continued its evolution to become a better language and to achieve convergence with the LOTOS NT language supported by the TRAIAN compiler. A major step was achieved in 2023 when version 3.11 of TRAIAN (and later, version 3.12) was integrated in the CADP distribution. Such integration brings two major benefits for the users of LNT:

- TRAIAN provides a full-fledged, native compiler front-end for LNT. TRAIAN is stricter than LNT2LOTOS, as it detects errors that LNT2LOTOS cannot catch because of its “lightweight” translation approach. For instance, TRAIAN does complete type checking, whereas LNT2LOTOS does not check types, simply generating LOTOS code and deferring type-checking duties to the LOTOS compilers CAESAR.ADT and CAESAR.

- The error and warning messages emitted by TRAIAN are easier to understand, since they refer to the LNT source code, while many messages of LNT2LOTOS refer to the generated LOTOS code.

The integration of TRAIAN in CADP is done by the LNT.OPEN shell script, which (unless it is called with the new “-notraian” option) invokes TRAIAN before invoking LNT2LOTOS. LNT.OPEN no longer invokes LNT_CHECK as the warning messages of TRAIAN about incomplete “case” statements are more informative than those of LNT_CHECK. If TRAIAN detects a fatal error, then LNT.OPEN stops and does not invoke LNT2LOTOS.

In 2023, the LNT language has been subject to numerous changes, among which:

- The syntax of LNT patterns was restricted: it is no longer allowed to write two (or more) consecutive unary operators without using parentheses.
- The usage of “with” clauses for LNT types was enhanced and made clearer, by permitting “with” clauses defining equality and (lexicographic) order relations for all types, permitting “with INSERT” clauses only for set and list types, restricting “with CARD” clauses to set types only, and allowing “with =” clauses (in addition to “with ==”) for all LNT types.
- The LNT_V1 library of predefined LNT types has been enhanced by adding eighteen new functions and renaming five other ones to make them more intuitive.
- The LNT2LOTOS Reference Manual and the LNT tools have been updated to document and implement these changes. The editor style files for LNT and many CADP demos have been also updated. The “upc” shell script was extended to ease the migration of LNT programs. Fourteen bugs have been fixed in the LNT tools.

To increase its interoperability with TRAIAN, the LNT2LOTOS translator was enhanced as follows:

- The grammar used for the syntax analysis of LNT2LOTOS was modified to be aligned with that of TRAIAN.
- Sets were implemented more efficiently, the worst-case complexity of set equality becoming linear instead of quadratic.
- LNT2LOTOS now supports field-selection and field-update expressions with event parameters, “with FIRST” and “with LAST” clauses in cascade types, as well as “with PRED” and “with SUCC” clauses in ranges, enumerations, and cascade types.
- The handling of warnings was improved by refining the dataflow analysis of the “for” loops in LNT functions to produce additional warnings (similar to the ones emitted by TRAIAN), displaying more precise line numbers in warning messages, and by no longer emitting warnings redundant with those of TRAIAN when LNT2LOTOS is invoked after TRAIAN (i.e., with its new “-traian” option).

7.1.2 Nested-Unit Petri Nets

Participants: Pierre Bouvier, Hubert Garavel.

Nested-Unit Petri Nets (NUPNs) is a model of computation that can be seen as an upward-compatible extension of P/T nets, which are enriched with structural information on their concurrent and hierarchical structure. Such structural information can easily be produced when NUPNs are generated from higher-level specifications (e.g., process calculi) and allows logarithmic reductions in the number of bits required to represent reachable states, thus enabling verification tools to perform better. For this reason, NUPNs have been so far implemented in thirteen verification tools developed in four countries, and adopted by two international competitions (the Model Checking Contest and the Rigorous Examination of Reactive Systems challenge). The complete theory of NUPNs is formalized in a journal article [3] and their PNML representation is described here.

In 2023, we continued our work on NUPNs, focusing on their connections with other formal models of concurrent systems, such as communicating automata, ordinary Petri nets, and process algebras. We proposed decompositions of NUPNs into communicating automata, decompositions of ordinary (safe) Petri nets into hierarchical NUPNs, and a translation from NUPNs to LNT. These results are presented in P. Bouvier’s PhD thesis [16].

7.1.3 Formal Modeling and Analysis of BPMN

Participants: Gwen Salaün *(correspondent)*, Ahang Zuo.

Business Process Model and Notation (BPMN) is a workflow-based notation that has been published as an ISO standard and has become the main language for business process modeling. Resource allocation is a critical problem in business processes due to the simultaneous execution of tasks and resource sharing among them. The number of allocated resources affects both the execution cost and time of the process. In the context of runtime processes, a well-defined resource allocation strategy is essential for optimising waiting times and costs by mitigating delays and enhancing resource utilisation.

In 2023, in the context of the MOAP project (see § 9.5.1), in collaboration with Yliès Falcone (CORSE project-team), we improved our prior work [24, 23] and proposed a generic approach for dynamically adjusting resource allocation when executing BPMN processes. The BPMN process is monitored in real-time, and the execution traces produced during its multiple executions are analysed. These execution traces are used to compute various properties or metrics of interest, including resource usage and average execution time. The approach then relies on predictive analytics to compute the future values of the aforementioned metrics. Based on these predicted results, strategies for the dynamic allocation of resources are defined, which anticipate changes in resource usage and thus dynamically update the number of resources in advance. This approach is fully automated using a toolchain and has been validated with multiple examples. This work led to a paper submitted to an international conference.

7.1.4 SYNTAX Compiler Generator

Participants: Pierre Boullier, Hubert Garavel, Frédéric Lang, Wendelin Serwe.

SYNTAX is a compiler-generation tool that generates lexical analyzers (scanners) and syntactic analyzers 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 TRAIAN and of most compilers of CADP.

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

In 2023, the development of SYNTAX has been particularly active in the following directions:

- The overall architecture of SYNTAX evolved, with a division of its code into four parts: “trunk”, which gathers tools for deterministic grammars (i.e., computer languages); “extensions”, which gathers tools for nondeterministic grammars (i.e., natural languages); and “outdated”, which gathers tools that are not longer maintained. The latter part was divided into two sub-parts: “oldies”, which contains tools that could be still of interest, and “deleted”, which archives abandoned tools.

- Significant cleanup was done, with deletion of duplicated files and archiving of obsolete software components.

- The quality of SYNTAX code improved by removing all warnings emitted by recent versions of the GCC and CLANG compilers. Many useless type casts have also been removed. Redundant macro-definitions (SXBOOLEAN, SXEXIT, SXFALSE, SXSHORT, SXTRUE, SXVOID, etc.) have been eliminated.

- SYNTAX was ported to 64-bit Windows and to recent versions of macOS. Support for obsolete architectures (Sparc, 32-bit Solaris, 32-bit macOS, etc.) has been dropped.

- The build system for SYNTAX, which was a complex mixture of scripts, makefiles, autogenesis/hypergenesis, and autotools has been dismantled, and merged into a unique script "sxmake" that centralises all operations. All SYNTAX tools have been simplified accordingly. In particular, their local hierarchy (“incl”, “src”, and “spec” directories) has been flattened, but "boot" directories and skeleton files have been created to highlight places where bootstrapping occurs.

- The LECL tool has been enhanced by P. Boullier, who introduced the notion of “zombie” lexical tokens to suppress warnings about unused tokens.

- The TABC tool, the name of which created confusion with another tool named TABLES_C, was renamed to SEMC.

- The documentation has been carefully enhanced and updated. The SYNTAX distribution has been enriched with published papers, presentation slides, and many README files based on explanations provided by P. Boullier.

- A new SXML library was added by H. Garavel, which helps generating directly an XML or JSON output from an input file, without building an (explicit) abstract syntax tree. This library has a reduced memory footprint due to the use of circular lists.

- The demo examples of SYNTAX have been enhanced. A new demo “simproc” has been added, which uses the LNT language and the TRAIAN compiler to build and traverse the abstract syntax tree. Three rarely used SYNTAX tools (pplecl, pprecor, and pptdef) have been turned into demo examples.

- The “lustre” demo has been enhanced. The grammar has been revised, to be closer to the official LUSTRE grammar. An syntax tree in XML is now produced as output, using the SEMC tool and the SXML library.

- The “f77” demo has been largely rewritten in the framework of a new collaboration with the RMOD/EVREF project-team (Lille). P. Boullier adapted the FORTRAN 77 grammar to pass the 192 official NIST tests. The grammar was also modified to factorize duplicated definitions, and to support "DO" loops without labels and loops terminated with “END DO”. The lexer and parser were modified to retain the comments present in FORTRAN 77 programs. The FORTRAN 77 pretty-printer was updated and made functional. On this basis, the RMOD/EVREF team started extending the grammar to generate an abstract tree in JSON, using the SEMC tool and the SXML library.

- H. Garavel, F. Lang, and W. Serwe gave various presentations of SYNTAX at ENS Ulm (Paris, France), LIG Grenoble, Saarland University (Germany), and Aix-les-Bains (France).

7.2 Parallel and Distributed Verification

7.2.1 Verification of Emergent Properties in Multi-agent Systems

Participants: Frédéric Lang.

Multi-agent systems are collections of autonomous components that interact with each other and with their shared environment. These systems may display collective properties that arise from the interplay between agents. Reasoning about these properties turns out to be hard, due to the very large state space that these systems usually exhibit. Therefore, automatic procedures to formally guarantee the emergence of such properties may prove helpful in the design of reliable artificial multi-agent systems.

In collaboration with Luca Di Stefano (Chalmers University, Sweden), we adapted the existing translation procedure from LAbS [37] to LNT, so that agents are now represented by networks of parallel LNT processes, enabling the application of compositional verification. We combined compositional verification with a static value analysis to prune the state space of individual agents and demonstrated the effectiveness of our approach by verifying a collection of representative systems. In 2023, this work led to a publication in an international conference [12].

7.3 Timed, Probabilistic, and Stochastic Extensions

7.3.1 Quantitative Analysis of BPMN Processes

Participants: Gwen Salaün *(correspondent)*, Quentin Nivon, Ahang Zuo.

Business process optimisation is a strategic activity in organisations because of its potential to increase profit margins and reduce operational costs. We consider business processes described using an extension of BPMN with quantitative aspects for modelling execution times and resources associated with tasks. A process is not executed once but multiple times, and multiple concurrent executions of a process compete for using the shared resources. In this context, it is particularly difficult to ensure correctness and efficiency of the multiple executions of a process.

In 2023, we followed two different research directions for analyzing quantitatively business processes:

- We considered process refactoring, a specific technique used for process optimisation, and we proposed a refactoring approach whose goal is to reduce the total execution time of a process and optimize the usage of the shared resources. This approach works by entirely refactoring the BPMN process in one step. First, a new version of the process is generated, which is as parallel as possible, while respecting the dependencies between tasks and gateways. Then, the resource usage of this process is computed. If some resources are overused, some tasks are put back in sequence to avoid possible delays involved by merge gateways. Finally, this optimised process is returned to the user. This process refactoring technique is fully automated by a tool that we implemented and applied on several examples for validation purposes. This work led to a publication in an international conference [14].
- In the context of the MOAP project (see § 9.5.1), in collaboration with Yliès Falcone (CORSE project-team), we improved our prior work [24], which relied on probabilistic model checking to automatically verify that multiple executions of a process respect some specific probabilistic property. This approach applies at runtime, thus the evaluation of the property is periodically carried out and the corresponding results updated. However, we go beyond runtime probabilistic analysis for BPMN, since we propose runtime enforcement techniques to keep executing the process while avoiding the violation of the property. We achieve this by monitoring techniques, computation of probabilistic models, probabilistic model checking, and runtime enforcement techniques. The approach has been implemented as a toolchain and its effectiveness has been validated on several realistic BPMN processes. This work led to a paper accepted for publication in an international conference.

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

7.4.1 Compositional Verification

Participants: Frédéric Lang.

The CADP toolbox contains various tools dedicated to compositional verification, among which EXP.OPEN, BCG_MIN, BCG_CMP, and SVL play a central role. EXP.OPEN explores on the fly the graph corresponding to a network of communicating automata (represented as a set of BCG files). BCG_MIN and BCG_CMP respectively minimize and compare behavior graphs modulo strong or branching bisimulation and their stochastic extensions. SVL (*Script Verification Language*) is both a high-level language for expressing complex verification scenarios and a compiler dedicated to this language.

In 2023, in addition to a bug fix for the macOS version of SVL, five enhancements have been brought: SVL now produces shorter log files that better take into account user interrupts and run-time errors; the error and warning messages issued by SVL have been aligned on those of other CADP compilers and now provide line numbers; better error or warning messages are emitted when reduction fails for some probabilistic or stochastic equivalence, and when both variables $DEFAULT_LOTOS_FILE and $DEFAULT_PROCESS_FILE are undefined; SVL only retries reduction using a stronger equivalence relation when this is possibly fruitful.

The BCG_MIN tool was improved so as to remove those transitions that became unreachable after applying probabilistic or stochastic equivalences (e.g., due to maximal progress).

In collaboration with Luca Di Stefano (Chalmers University, Sweden), we extended sharp bisimulation to the framework of systems with priorities (with an application to multi-agent systems). This work led to a publication in an international journal [10].

7.4.2 On-the-fly Resolution of Boolean Equation Systems

Participants: Hubert Garavel, Radu Mateescu.

OPEN/CAESAR is an extensible, modular, language-independent software framework for exploring implicit graphs (i.e., defined by their *post* function, which enumerates the transitions going out of each vertex). This key component of CADP is used to build simulation, execution, verification, and test generation tools.

CAESAR_SOLVE_1 is a generic software library, based on OPEN/CAESAR, for solving Boolean Equation Systems (BESs) of alternation depth 1 (i.e., without mutual recursion between minimal and maximal fixed point equations) on the fly. This library is at the core of several CADP verification tools, namely the equivalence checker BISIMULATOR, the minimization tool REDUCTOR, and the model checker EVALUATOR. The resolution method is based on Boolean graphs, which provide an intuitive representation of dependencies between Boolean variables, and which are handled implicitly, in a way similar to the OPEN/CAESAR interface [26].

In 2023, a data base of 10,500+ BESs has been prepared, on which the CAESAR_SOLVE_1 library and the BES_SOLVE tool have been tested systematically. This effort led to six bug fixes: the "unique" and "mode N" pragmas contained in BES files were ignored; the "-parallel" option of BES_SOLVE could fail if a non-writable BES_SOLVE binary was already present on some remote machine; the resolution algorithm A8 could stop with a "memory shortage" error; the resolution algorithm A9 could halt with a segmentation fault; the resolution algorithms A8, A10, and A11 could generate incorrect diagnostics. Also, the memory used for resolution was reduced by a factor between 10% and 50% by better dimensioning the internal table that stores Boolean variables.

7.4.3 Runtime Enforcement for Adaptive Industrial Control Systems

Participants: Irman Faqrizal, Gwen Salaün *(correspondent)*.

Adaptive industrial control systems can reliably adapt to specific requirements with minimal effort. IEC 61499 is a promising standard that allows downtimeless system evolution such that an application can be modified at runtime to satisfy the requirements. However, an IEC 61499 application consisting of multiple Function Blocks (FBs) can be modified in many different ways, such as inserting or deleting FBs, creating new FBs with their respective internal behaviours, and adjusting the connections between FBs. These changes require considerable effort and cost, and there is no guarantee to satisfy the requirements.

In 2023, in the framework of the D-IIoT project (see § 9.5.2), in collaboration with Yliès Falcone (CORSE project-team), we applied runtime enforcement techniques to support adaptive IEC 61499 applications. This set of techniques can modify the runtime behaviour of a system according to specific requirements. Our approach begins with specifying the requirements as a state machine-based notation called contract automaton. This contract automaton is then used to synthesise an enforcer as an FB. Finally, the new FB is integrated into the application to execute according to the requirements. A tool support has been developed to automate the approach. In addition, experiments were performed to evaluate the performance of enforcers by measuring the execution time of several applications before and after the integration of enforcers. This work led to a paper submitted to an international journal.

7.4.4 Other Component Developments

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

In 2023, in addition to seven bug fixes in CAESAR (invoked with its "-graph" option), EUCALYPTUS, EVALUATOR 4 and 5 (on Solaris architecture only), EXP.OPEN (on large composition expressions containing priority operators), MCL_EXPAND (when invoking EVALUATOR 5 with its "-acyclic" option on probabilistic formulas), and XTL (when manipulating label fields of type String), various enhancements have been brought to the following CADP tools:

- The BCG monitor was made much faster when generating labelled transition systems containing many different labels.
- BCG_MIN (unless when called with "-class" option) now removes, before undertaking probabilistic or stochastic minimization, all states and transitions that are unreachable due to the "maximal progress" assumption.
- XTL no longer emits warnings on Linux when evaluating an XTL program on a BCG file containing no transitions.
- The two functions ADT_GCD_NAT() and ADT_SCM_NAT() of the natural number library now use faster algorithms that are less prone to numeric overflow.
- In CADP, better file compression is now achieved using "gzip" instead of "compress".
- The memory efficiency of demo 16 was enhanced by adding a "!card" type pragma. The LOTOS code of demos 23 and 25 was shortened, without loss of functionality, by 42% (from 2090 down to 1208 non-blank lines) and 27% (from 767 down to 560 non-blank lines), respectively. Also, demos 23, 24, and 25 have been translated from LOTOS to LNT, still preserving strong bisimilarity.

The evolutions of operating systems and C compilers dictated many changes in the CADP infrastructure, among which:

- Update of shell scripts to use modern PostScript viewers (e.g., Atril and Evince) and to avoid warnings emitted by recent versions of GNU grep.
- Update of shell scripts and documentation to support Solaris 11 and recent versions of Oracle's C compiler; with these changes, CADP properly runs on Solaris 11.4.
- Updates of scripts, include files, and documentation to avoid warnings emitted by the recent versions of Clang and to support macOS 14 "Sonoma", Xcode version 14.3, and recent versions of XQuartz.
- Complete port of CADP to 64-bit Windows, with an upgrade of TCL/TK/TIX to their latest versions and subsequent adaptation of the CADP graphical tools (XSIMULATOR, OCIS); progressive migration from 32-bit to 64-bit Windows executables, the latter eventually replacing the former since they are faster; support of Windows 11 and the most recent versions of Cygwin.

7.5 Real-Life Applications and Case Studies

7.5.1 Autonomous Car

Participants: Jean-Baptiste Horel, Philippe Ledent, Radu Mateescu *(correspondent)*,
Lucie Muller, Wendelin Serwe.

A common practice to evaluate autonomous vehicles is simulation, which requires to specify realistic scenarios, in particular critical ones, occurring rarely and potentially dangerous to reproduce on the road. Such scenarios may be either generated randomly, or specified manually. Randomly generating scenarios is easy, but their relevance might be difficult to assess. Manually specified scenarios can focus on a given feature, but their design might be difficult and time-consuming, especially to achieve satisfactory coverage.

In the framework of the ArchitectECA2030 (see § 9.3.2) and PRISSMA (see § 9.4.1) projects and in collaboration with Lina Marsso (University of Toronto, Canada), Christian Laugier, Anshul Paigwar, 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 [34], 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 2023, we extended this approach with a statistical analysis to assess the collision-risk estimation of a perception component. We applied the approach to analyze more complex scenarios, with a non-linear behavior of vehicle and obstacles (varying speeds and curved trajectories), in order to evaluate the output of the perception component in more realistic situations. These results have been published in Lucie Muller's PhD thesis [36] and in an international journal [11].

7.5.2 Manufacturing Application

Participants: Irman Faqrizal, Gwen Salaün *(correspondent)*.

The ever-increasing complexity of industrial control systems generates a demand for reliable development methods. IEC 61499, a recent industrial standard, helps to develop complex distributed systems based on their positive characteristics, namely reusability, reconfigurability, interoperability, and portability. Formal verification techniques, such as model checking, have been proposed to ensure the correctness of these systems during the design time. However, they do not consider the presence of the environment that can impact the application behaviour at runtime.

In 2023, in collaboration with Tatiana Liakh, Valeriy Vyatkin, and Midhun Xavier (Lulea University of Technology, Sweden), we applied probabilistic model checking on an IEC 61499-based manufacturing application. We devised several probabilistic properties to be checked. The results were visualised graphically to be analyzed, which allows one to optimise the system's quantitative features, such as productivity. This work led to a paper accepted for publication in an international conference.

8 Bilateral contracts and grants with industry

8.1 Start-up Project

Participants: Ajay Muroor-Nadumane, Gwen Salaün *(correspondent)*.

G. Salaün and A. Muroor-Nadumane initiated the Voyance Systems start-up, which became part of Inria Startup Studio in February 2022 and continued its activity in 2023. Voyance Systems provides integrated analytics enabling enterprises to improve efficiency and reduce greenhouse gas emissions.

Voyance Systems leverages model-based techniques to analyse different types of processes, ranging from operations workflows to complex processes in retail, manufacturing, logistics, and energy sectors. The analyses enable technical analysts and business leaders to optimize their workflows in terms of resource utilization, greenhouse gas emissions, cost, throughput, and other key performance indicators.

8.2 Bilateral grants with industry

8.2.1 ST Microelectronics

Participants: Philippe Ledent, Radu Mateescu *(correspondent)*, Wendelin Serwe.

Ph. Ledent is supported by a CIFRE PhD grant (from October 2020 to September 2023) from ST Microelectronics (Grenoble) on the formal validation of security requirements for Systems-on-Chip, under the supervision of Hajer Ferjani (ST Microelectronics), Radu Mateescu (CONVECS), and Wendelin Serwe (CONVECS).

9 Partnerships and cooperations

9.1 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.

9.1.1 Other international collaborations

In 2023, we had scientific relations with several universities and institutes abroad, including:

- University of Málaga, Spain (Francisco Durán),
- Saarland University, Germany (Holger Hermanns),
- University of Urbino, Italy (Marco Bernardo),
- University of Toronto, Canada (Lina Marsso),
- Lulea University of Technology, Sweden (Tatiana Liakh, Valeriy Vyatkin, and Midhun Xavier),
- Chalmers University, Sweden (Luca Di Stefano).

9.2 International research visitors

9.2.1 Visits of international scientists

- Lina Marsso (University of Toronto, Canada) visited our team on May 9, 2023. She gave a talk entitled "*Early Verification of Legal Compliance via Bounded Satisfiability Checking*".
- Marco Bernardo (University of Urbino, Italy) visited our team on August 24–25, 2023. He gave on August 25, 2023 a talk entitled "*A Process Algebraic Theory of Reversible Systems*" on August 25, 2023.

9.2.2 Visits to international teams

Research stays abroad

- G. Salaün visited the University of Málaga (Spain) from May 27 to June 10, 2023.

9.3 European initiatives

9.3.1 Horizon Europe

A-IQ Ready

Participants: Frédéric Lang, Radu Mateescu *(correspondent)*, Wendelin Serwe.

A-IQ Ready project on 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 FUER LANDESVERTEIDIGUNG (BMLV), 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
- SYNOPSYS 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
- CARGOTEC 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 AG, Austria
- TTTECH COMPUTERTECHNIK AG, Austria
- ELEKTRONIKAS UN DATORZINATNU INSTITUTS (EDI), Latvia
- 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.

9.3.2 H2020 projects

ArchitectECA2030

Participants: Radu Mateescu *(correspondent)*, Lucie Muller, Wendelin Serwe.

ArchitectECA2030 project on cordis.europa.eu

Title: Trustable architectures with acceptable residual risk for the electric, connected and automated cars

Duration: From July 1, 2020 to December 31, 2023

Partners:

- UAB TERAGLOBUS, Lithuania
- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE (INRIA), France
- TECHNISCHE UNIVERSITAET GRAZ (TU GRAZ), Austria
- INFINEON TECHNOLOGIES AUSTRIA AG (IFAT), Austria
- BOARD OF REGENTS OF NEVADA SYSTEM OF HIGHER EDUCATION, United States
- SafeTRANS e.V. (SafeTRANS), Germany
- TRACSENSE AS, Norway
- NXTECH AS, Norway
- INSTITUT MIKROELEKTRONICKYCH APLIKACI SRO (IMA), Czechia
- INFINEON TECHNOLOGIES AG (IFAG), Germany
- VYSOKE UCENI TECHNICKE V BRNE (BRNO UNIVERSITY OF TECHNOLOGY), Czechia
- SINTEF AS (SINTEF), Norway
- SMARTSOL SIA, Latvia
- VIRTUAL VEHICLE RESEARCH GMBH (VIF), Austria
- NXP SEMICONDUCTORS NETHERLANDS BV, Netherlands
- AVL LIST GMBH (AVL), Austria
- DATASOFT EMBEDDED GMBH (DATASOFT EMBEDDED), Austria
- SBA RESEARCH GEMEINNUTZIGE GMBH (SBA), Austria
- VOLKSWAGEN AKTIENGESELLSCHAFT (VW AG), Germany
- TECHNISCHE UNIVERSITEIT DELFT (TU Delft), Netherlands
- TECHNISCHE UNIVERSITAET DRESDEN (TUD), Germany

Inria contact: Radu Mateescu

Coordinator: Georg Stettinger (IFAG)

Summary: Independent validation is fundamental to emphasise the capability and safety of any solution in the electric, connected and automated (ECA) vehicles space. It is vital that appropriate and audited testing takes place in a controlled environment before any deployment takes place. As the software and hardware components come from multiple vendors and integrate in numerous ways, the various levels of validation required must be fully understood and integration with primary and secondary parts must be considered.

The key targets of ArchitectECA2030 are the robust mission-validated traceable design of electronic components and systems (ECS), the quantification of an accepted residual risk of ECS for ECA vehicles to enable type approval, and an increased end-user acceptance due to more reliable and robust ECS. The proposed methods include automatic built-in safety measures in the electronic circuit design, accelerated testing, residual risk quantification, virtual validation, and multi-physical and stochastic simulations.

The project will implement a unique in-vehicle monitoring device able to measure the health status and degradation of the functional electronics empowering model-based safety prediction, fault diagnosis, and anomaly detection. A validation framework comprised of harmonized methods and tools able to handle quantification of residual risks using data different sources (e.g. monitoring devices, sensor/actuators, fleet observations) is provided to ultimately design safe, secure, and reliable ECA vehicles with a well-defined, quantified, and acceptable residual risk across all ECS levels. The project brings together stakeholders from ECS industry, standardization and certification bodies (e.g. ISO, NIST, TUEV), test field operators, insurance companies, and academia closely interacting with ECSEL lighthouse initiative Mobility.E to align and influence emerging standards and validation procedures for ECA vehicles.

The main contributions of CONVECS in the project are the formal modeling and validation of components embedded in autonomous vehicles.

9.3.3 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.

9.4 National initiatives

9.4.1 Fonds pour l'innovation et l'industrie

PRISSMA

Participants: Jean-Baptiste Horel, Radu Mateescu *(correspondent)*.

PRISSMA is a project funded by the *Fonds pour l'innovation et l'industrie* within the *Grand défi 2 : sécuriser, certifier et fiabiliser les systèmes fondés sur l'intelligence artificielle* programme. The project involves 19 industrial partners (among which ANSYS, RATP, and VALEO), as well as Université Gustave-Eiffel, LNE, and Inria (project-teams CHROMA and CONVECS). PRISSMA aims at proposing a platform enabling to release the technological locks that hamper the deployment of secure IA-based systems and to integrate all the necessary elements for the homologation activities of autonomous vehicles and their validation in real environments given by use cases.

PRISSMA started in April 2021 for three years. The main contributions of CONVECS to PRISSMA are the formal modeling and validation of perception components of the autonomous vehicle.

9.4.2 Other national collaborations

We had sustained scientific relations with the following researchers:

- Nicolas Amat and Silvano Dal Zilio (LAAS-CNRS, Toulouse),
- Pierre Boullier (formerly ALPAGE project-team, Paris),
- Stéphane Ducasse, Nicolas Anquetil, and Larisa Safina (RMOD/EVREF project-team, Lille).

9.5 Regional initiatives

9.5.1 Pack ambition recherche région Auvergne-Rhône-Alpes

MOAP

Participants: Gwen Salaün *(correspondent)*, Ahang Zuo.

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.

9.5.2 Persyval Labex

D-IIoT

Participants: Irman Faqrizal, Gwen Salaün *(correspondent)*.

D-IIoT is a project funded by the Persyval Labex via the ANR ("*Agence Nationale de la Recherche*"). The project involves research teams of three local laboratories (CEA, LIG, VERIMAG). D-IIoT aims at studying and proposing new techniques to support the execution of long-running and evolving IIoT (Industrial IoT) applications with dependability guarantees (e.g., security, correctness).

D-IIoT started in October 2021 for three years. The main contributions of CONVECS to D-IIoT are the formal modeling, verification and reconfiguration of IIoT applications.

10 Dissemination

Participants: Pierre Bouvier, Irman Faqrizal, Hubert Garavel, Frédéric Lang, Radu Mateescu, Quentin Nivon, Gwen Salaün, Wendelin Serwe, Ahang Zuo.

10.1 Promoting scientific activities

10.1.1 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 32 models, among which 10 were deposited by CONVECS.
- P. Bouvier and H. Garavel are members 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 FOCLASA (*International Workshop on Foundations of Coordination Languages and Self-Adaptative Systems*) workshop series since 2011.
- 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.

10.1.2 Scientific events: selection

Member of the conference program committees

- H. Garavel was a programme committee member of FMICS'2023 (*28th International Conference on Formal Methods for Industrial Critical Systems*), Antwerp, Belgium, September 18-23, 2023.
- H. Garavel was a programme committee member of SEFM'2023 (*21st International Conference on Software Engineering and Formal Methods*), Eindhoven, The Netherlands, November 6-10, 2023.
- G. Salaün was a programme committee member of DOORS'2023 (*3rd Edge Computing Workshop*), Zhytomyr, Ukraine, April 7, 2023.
- G. Salaün was a programme committee member of FASE'2023 (*26th International Conference on Fundamental Approaches to Software Engineering*), Paris, France, April 24-27, 2023.
- G. Salaün was a programme committee member of ENASE'2023 (*18th International Conference on Evaluation of Novel Approaches to Software Engineering*), Prague, Czech Republic, April 24-25, 2023.
- G. Salaün was a programme committee member of SEAMS'2023 (*18th Symposium on Software Engineering for Adaptive and Self-Managing Systems*), Melbourne, Australia, May 15-16, 2023.
- G. Salaün was a programme committee member of ICECCS'2023 (*27th International Conference on Engineering of Complex Computer Systems*), Toulouse, France, June 14-16, 2023.
- G. Salaün was a programme committee member of COMPSAC-SETA'2023 (*IEEE International Conference on Computers, Software, and Applications - Software Engineering Technologies and Applications*), Torino, Italy, June 27-29, 2023.
- G. Salaün was a programme committee member of ICSOFT'2023 (*18th International Conference on Software Technologies*), Rome, Italy, July 10-12, 2023.
- G. Salaün was a programme committee member of FM-BPM'2023 (*1st International Workshop on Formal Methods for Business Process Management*), Utrecht, The Netherlands, September 11, 2023.
- G. Salaün was a programme committee member of FACS'2023 (*19th International Conference on Formal Aspects of Component Software*), virtual event, October 19-20, 2023.
- G. Salaün was a programme committee member of SEFM'2023.
- G. Salaün was a programme committee member of ICTAC'2023 (*20th International Colloquium on Theoretical Aspects of Computing*), Lima, Peru, December 4-8, 2023.

Reviewer

- P. Bouvier was a reviewer for TOOLympics@ETAPS'2023 (*Competitions on Formal Methods*).
- F. Lang was a reviewer for FASE'2023, ICALP'2023 (*50th EATCS International Colloquium on Automata, Languages and Programming*), MFCS'2023 (*48th International Symposium on Mathematical Foundations of Computer Science*), and SEFM'2023.
- R. Mateescu was a reviewer for SPIN'2023 (*29th International Symposium on Model Checking Software*).
- W. Serwe was a reviewer for ESORICS'2023 (*28th European Symposium on Research in Computer Security*), ICTAC'2023, and SEFM'2023.
- A. Zuo was a reviewer for ICSOFT'2023.

10.1.3 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

- Frédéric Lang was a reviewer for FAC (*Formal Aspects of Computing*) and STVR (*Software Testing, Verification, and Validation*).
- Wendelin Serwe was a reviewer for SCP (*Science of Computer Programming*).

10.1.4 Software Dissemination and Internet Visibility

The CONVECS project-team distributes several software tools, among which the CADP toolbox.

In 2023, the main facts are the following:

- We prepared and distributed twelve successive versions (2023-a to 2023-l) of CADP.
- We granted CADP licenses for 145 different computers in the world.

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

By the end of December 2023, the CADP forum, opened in 2007 for discussions regarding the CADP toolbox, had over 481 registered users and over 1975 messages had been exchanged.<br>

Other teams used the CADP toolbox for various case studies:

- Vulnerability identification of operational technology protocols [17]
- Using process algebra for validating UML statecharts [22]
- Verification of circuits for systolic array parallel computation [19]

10.1.5 Invited talks

- H. Garavel and F. Lang participated to the scientific workshop ProgLang@Inria (ENS Ulm, Paris, January 8-9, 2023). H. Garavel gave a talk entitled "*Introduction au système SYNTAX pour la génération de compilateurs et de traducteurs*". F. Lang gave a talk entitled "*Construction de compilateurs en utilisant SYNTAX et LNT*".
- H. Garavel and W. Serwe participated to the workshop organized by the MFML (*Méthodes Formelles, Modèles et Langages*) axis of the LIG laboratory (Grenoble) on May 11, 2023. H. Garavel gave a talk entitled "*Introduction au système SYNTAX pour la génération de compilateurs et de traducteurs*". W. Serwe gave a talk entitled "*Compiler Construction using SYNTAX and LNT*".

- H. Garavel participated in Open Problems in Concurrency Theory OPCT (Bertinoro, Italy) on June 26–30, 2023. On June 26, 2023, he gave a lecture entitled "*What Was Wrong with Process Calculi – and How To Recover?*".

- H. Garavel visited the DEPEND group led by Holger Hermanns (Saarland University, Germany) from August 28 to September 1, 2023. On August 31, 2023, he gave a lecture entitled "*A Simple Approach for Building Compiler Front-ends*".

- G. Salaün participated to the VELVET days workshop organized by the GDR GPL (Nantes) on December 13-14, 2023. On December 14, 2023, he gave a talk entitled "*Automated Verification of TOSCA Workflows*".

10.1.6 Scientific expertise

- G. Salaün was president of the CES25 evaluation committee of ANR, domain *Sciences et Technologies Logicielles - Réseaux, Infrastructures, Calcul Haute Performance*.

- G. Salaün was member of the expert committee for the HCERES evaluation of the laboratory FEMTO-ST (*Franche-Comté Électronique, Mécanique, Thermique et Optique – Sciences et Technologies*, UMR CNRS 6174) held on January 17–19, 2023.

10.1.7 Research administration

- Until July 2023, F. Lang was 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 succeeded him on this date.

- R. Mateescu is the scientific correspondent of the International Partnerships for Inria Grenoble.

- R. Mateescu is a member of the "*Comité d'Orientation Scientifique*" for Inria Grenoble. In 2021, he also participated to the recruitment jury for CRCN (*Chargé de Recherche de Classe Normale*) and ISFP (Inria Starting Faculty Positions) at Inria Grenoble.

- R. Mateescu is representative of Inria Grenoble at the International Relations and Outreach of Université Grenoble Alpes (UGA).

- R. Mateescu is member of the council of the Mathematics, Information and Communication Sciences (MSTIC) research department of UGA.

- G. Salaün is the director of the MSTIC research department of UGA since September 2023.

- G. Salaün is the head of the *Métiers du Multimédia et de l'Internet* (MMI) department at IUT1/UGA.

- G. Salaün is a member of the administration council of the IUT1/UGA.

- G. Salaün is a member of the Scientific Committee of the PCS (*Pervasive Computing Systems*) action of the PERSYVAL Labex.

- G. Salaün is a member of the council of the LIG laboratory.

- W. Serwe is a member of the "*Comité de Centre*" at Inria Grenoble.

10.2 Teaching - Supervision - Juries

10.2.1 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 2023, we carried out the following teaching activities:

- I. Faqrizal gave courses on “*Introduction to Node.js*” (32 hours “*équivalent TD*”) to L3 students of IUT1/UGA.
- H. Garavel gave lectures on “*Probabilistic Models, Stochastic Models, and Static/Dynamic Fault Trees*” (12 hours “*équivalent TD*” as part of a course on “*System Design: Real-Time, Stochastic, and Analog/Digital*” to second year students of the MOSIG.
- H. Garavel was a jury member for the MOSIG master defenses (June and September 2023).
- 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 and second year students of the MOSIG.
- Q. Nivon participated on lectures on “*Management of Relational Data and Applications*” (39 hours “*équivalent TD*”) to L2 students of UGA and on “*Semantics of Programming Languages and Compilation*” (30 hours “*équivalent TD*”) to students of the Master 1 Informatique of UGA.
- G. Salaün taught about 240 hours of classes (algorithmics, Web development, object-oriented programming) at the MMI department of IUT1/UGA. He is also headmaster of the “*Services Mobiles et Interface Nomade*” (SMIN) professional licence (third year of university) at IUT1/UGA, and co-headmaster of the “*alternance*” (apprenticeship) at the MMI department.
- 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 2023.
- A. Zuo gave courses on “*Data Extraction*” (21 hours “*équivalent TD*”) to L3 (License Pro) students of IUT2/UGA and on “*Basics of software development: modularisation and testing*” (67 hours “*équivalent TD*”) to L2 students of UGA.

10.2.2 Supervision

- PhD: P. Bouvier, “*Systèmes concurrents hiérarchiques :équivalence, analyse et structuration*”, Université Grenoble Alpes, defended on October 12, 2023, H. Garavel and R. Mateescu
- PhD: L. Muller, “*Modélisation formelle et validation pour des véhicules automatisés*”, Université Grenoble Alpes, defended on December 15, 2023, R. Mateescu and W. Serwe
- PhD in progress: I. Faqrizal, “*Monitoring and Deployment of IIoT Applications*”, Université Grenoble Alpes, since October 2021, G. Salaün and Yliès Falcone
- 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
- PhD in progress: P. Ledent, “*Formal Validation of Security Requirements for a System-on-Chip Architecture*”, Université Grenoble Alpes, since October 2020, R. Mateescu, W. Serwe, and Hajer Ferjani
- PhD in progress: Q. Nivon, “*Analyse, optimisation et debugging de processus BPMN*”, Université Grenoble Alpes, since October 2022, G. Salaün
- PhD in progress: Choukri Soueidi, “*Instrumentation expressive et correcte de programmes distribués et vérification à l’exécution*”, Université Grenoble Alpes, since October 2020, G. Salaün and Y. Falcone
- PhD in progress: A. Zuo, “*Modelling, Optimization and Predictive Analysis of Business Processes*”, Université Grenoble Alpes, since October 2020, G. Salaün and Y. Falcone

10.2.3 Juries

- G. Salaün was jury president for Léo Gourdin's PhD thesis, entitled "*Validation formelle de transformations intra-procédurales par simulation symbolique défensive*", defended at UGA on December 12, 2023.
- G. Salaün was jury president for Asfand Yar's PhD thesis, entitled "*An xDSL-Based Framework for Validation of Railway Models: Application to ERTMS/ETCS and EULYNX*", defended at UGA on December 19, 2023.
- G. Salaün was jury president for Diego Diaz's PhD thesis, entitled "*Process Performance Indicators Variability integrated to Customizable Process Models*", defended at UGA on December 20, 2023.
- G. Salaün was jury member for Rabéa Ameur-Boulifa's Habilitation thesis, entitled "*Contributions to the Design of Safe Complex Systems*", defended at Université Côte d'Azur on January 27, 2023.

11 Scientific production

11.1 Major publications

[1] 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. URL: https://hal.inria.fr/hal-01290465.

[2] 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. URL: https://hal.inria.fr/hal-01412911.

[3] H. Garavel. '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. URL: https://hal.inria.fr/hal-02072190.

[4] H. Garavel, 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. URL: https://hal.inria.fr/hal-01247507.

[5] H. Garavel, 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. URL: http://hal.inria.fr/hal-00715056.

[6] H. Garavel, 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. URL: https://hal.inria.fr/hal-01621670.

[7] 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. URL: https://hal.inria.fr/hal-01920273.

[8] 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. URL: http://hal.inria.fr/hal-00671321.

11.2 Publications of the year

International journals

[9] N. Amat, P. Bouvier and H. Garavel. 'A Toolchain to Compute Concurrent Places of Petri Nets'. In: *LNCS Transactions on Petri Nets and Other Models of Concurrency*. Lecture Notes in Computer Science 14150 (1st Nov. 2023), pp. 1–26. DOI: 10.1007/978-3-662-68191-6_1. URL: https://inria.hal.science/hal-04392784.

[10] L. Di Stefano and F. Lang. 'Compositional verification of priority systems using sharp bisimulation'. In: *Formal Methods in System Design* (17th May 2023). DOI: 10.1007/s10703-023-00422-1. URL: https://inria.hal.science/hal-04103681.

[11] J.-B. Horel, P. Ledent, L. Marsso, L. Muller, C. Laugier, R. Mateescu, A. Paigwar, A. Renzaglia and W. Serwe. 'Verifying Collision Risk Estimation using Autonomous Driving Scenarios Derived from a Formal Model'. In: *Journal of Intelligent and Robotic Systems* 107.4 (Apr. 2023), pp. 1–45. DOI: 10.1007/s10846-023-01808-3. URL: https://inria.hal.science/hal-04138579.

International peer-reviewed conferences

[12] L. Di Stefano and F. Lang. 'Compositional Verification of Stigmergic Collective Systems'. In: VMCAI 2023 - 24th International Conference on Verification, Model Checking , and Abstract Interpretation. Boston, United States, 16th Jan. 2023, pp. 1–22. URL: https://inria.hal.science/hal-03869922.

[13] J.-B. Horel, R. Baruffa, L. Rummelhard, A. Renzaglia and C. Laugier. 'A Navigation-Based Evaluation Metric for Probabilistic Occupancy Grids: Pathfinding Cost Mean Squared Error'. In: *IEEE International Conference on Intelligent Transportation Systems*. ITCS 2023 - 26th IEEE International Conference on Intelligent Transportation Systems. Bilbao, Spain, Spain: IEEE, 2023, pp. 1–6. URL: https://hal.science/hal-04211125.

[14] Q. Nivon and G. Salaün. 'Refactoring of Multi-instance BPMN Processes with Time and Resources'. In: *Lecture Notes in Computer Science*. SEFM 2023 - International Conference on Software Engineering and Formal Methods. Vol. 14323. Lecture Notes in Computer Science. Eindhoven, Netherlands: Springer Nature Switzerland, 31st Oct. 2023, pp. 226–245. DOI: 10.1007/978-3-031-47115-5_13. URL: https://inria.hal.science/hal-04278929.

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

[15] *Editorial for FACS 2021 special section (SoSyM)* 22.2 (1st Feb. 2023). DOI: 10.1007/s10270-023-01088-3. URL: https://inria.hal.science/hal-04310133.

Doctoral dissertations and habilitation theses

[16] P. Bouvier. 'Hierarchical concurrent systems : equivalence, analysis and structuring'. Université Grenoble Alpes [2020-....], 12th Oct. 2023. URL: https://theses.hal.science/tel-04412686.

11.3 Cited publications

[17] M. Boeding, M. Hempel and H. Sharif. 'Vulnerability Identification of Operational Technology Protocol Specifications Through Formal Modeling'. In: *Proceedings of the 16th International Conference on Signal Processing and Communication System (ICSPCS'2023)*. IEEE, 2023, pp. 1–6. DOI: 10.1109/ICSPCS58109.2023.10261127.

[18] 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.

[19] Y. Chiba and K. Wasaki. 'Description and Verification of Systolic Array Parallel Computation Model in Synchronous Circuit Using LOTOS'. In: *Proceedings of the 20th International Conference on Information Technology-New Generations (ITNG'2023)*. Ed. by S. Latifi. Vol. 1445. Advances in Intelligent Systems and Computing. Springer, 2023, pp. 379–386.

[20] 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.

[21] 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.

[22] S. Doostali, S. M. Babamir and M. Javani. 'Using a Process Algebra Interface for Verification and Validation of UML Statecharts'. In: *Computer Standards & Interfaces* 86 (2023), p. 103739. DOI: https://doi.org/10.1016/j.csi.2023.103739.

[23] F. Durán, Y. Falcone, C. Rocha, G. Salaün and A. Zuo. 'From Static to Dynamic Analysis and Allocation of Resources for BPMN Processes'. In: *WRLA 2022 - 14th International Workshop on Rewriting Logic and its Applications*. Munich, Germany, Apr. 2022, pp. 1–18. DOI: 10.1007/978-3-031-12441-9_1. URL: https://inria.hal.science/hal-03766148.

[24] Y. Falcone, G. Salaün and A. Zuo. 'Probabilistic Model Checking of BPMN Processes at Runtime'. In: *iFM 2022 - International Conference on integrated Formal Methods*. Lugano, Switzerland, June 2022, pp. 1–17. DOI: 10.1007/978-3-031-07727-2_11. URL: https://inria.hal.science/hal-03665305.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] H. Garavel and W. Serwe. 'State Space Reduction for Process Algebra Specifications'. In: *Theoretical Computer Science* 351.2 (Feb. 2006), pp. 131–145.

[31] 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.

[32] M. Hennessy and R. Milner. 'Algebraic Laws for Nondeterminism and Concurrency'. In: *Journal of the ACM* 32 (1985), pp. 137–161.

[33] J. Magee and J. Kramer. *Concurrency: State Models and Java Programs*. 2006th ed. Wiley, Apr. 2006.

[34] 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. URL: https://hal.inria.fr/hal-01777861.

[35] 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.

[36] L. Muller. ‘Modélisation formelle et validation pour des véhicules automatisés’. PhD Thesis. Université Grenoble Alpes, Dec. 2023.

[37] R. D. Nicola, L. Di Stefano and O. Inverso. ‘Multi-agent Systems with Virtual Stigmergy’. In: *Sci. Comput. Program.* 187 (2020), p. 102345.