IN PARTNERSHIP WITH:
CNRS

Institut polytechnique de Grenoble

Université Joseph Fourier (Grenoble)

Activity Report 2012

Team CONVECS

Construction of verified concurrent systems

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

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Embedded and Real Time Systems

Table of contents

Team CONVECS

Keywords: Formal Methods, Model-checking, Verification, Distributed Algorithms, Markovian Model

Creation of the Team: January 01, 2012 .

1. Members

Research Scientists
Hubert Garavel [Senior Researcher Inria, part-time]
Radu Mateescu [Team leader, Senior Researcher Inria, HdR]
Frédéric Lang [Researcher Inria]
Wendelin Serwe [Researcher Inria]

Faculty Member
Gwen Salaün [Grenoble INP, HdR]

Engineer
Eric Léo [since September 3, 2012]

PhD Students
Rim Abid [with ERODS team of LIG, since October 3, 2012]
Hugues Evrard [since March 15, 2012]
Fatma Jebali [since December 3, 2012]
Abderahman Kriouile [CIFRE with STMicroelectronics, since March 12, 2012]
Raquel Oliveira [with IIHM team of LIG, since October 15, 2012]

Post-Doctoral Fellows
Matthias Güdemann [until October 16, 2012]
Jingyan Jourdan-Lu [since December 3, 2012]
Lina Ye [since November 5, 2012]

Administrative Assistant
Helen Pouchot [shared with another team]

Other
Alexandre Dumont [internship from January 31 to May 22, 2012]

2. Overall Objectives

2.1. Overview

The CONVECS 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.

2.2. Highlights of the Year

- F. Lang and R. Mateescu's paper entitled "*Partial Model Checking Using Networks of Labeled Transition Systems and Boolean Equation Systems*" [15] was selected as one among the three "*best paper nominees*" of the TACAS 2012 conference, which had 36 papers published out of 147 submitted.
- At the end of 2012, the number of software licenses granted for the CADP toolbox since the beginning of its distribution has reached 10000.

3. Scientific Foundations

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 [36] 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*) [56], 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 analyses 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 operability 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 analyses.

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 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 [42], 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 enable 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 catalogue of interoperable components serving as building blocks for new analysis features. A long-term goal of this approach is to provide an increasingly large catalogue 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 catalogue. 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

4.1. Application Domains

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

- *Bioinformatics:* genetic regulatory networks, nutritional stress response, metabolic pathways,
- *Consumer electronics:* home networking, video on-demand,
- *Databases:* transaction protocols, distributed knowledge bases, stock management,
- *Distributed systems:* virtual shared memory, distributed file systems, election algorithms, dynamic reconfiguration algorithms, fault tolerance algorithms, cloud computing,
- *Embedded systems:* smart-card applications, air traffic control, avionic systems,
- *Hardware architectures:* asynchronous circuits, multiprocessor architectures, systems on chip, networks on chip, bus arbitration protocols, 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. Software

5.1. The CADP Toolbox

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

We maintain and enhance CADP (*Construction and Analysis of Distributed Processes* – formerly known as *CAESAR/ALDEBARAN Development Package*) [4], a toolbox for protocols and distributed systems engineering (see http://cadp.inria.fr). In this toolbox, we develop and maintain the following tools:

- CAESAR.ADT [41] 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 [48], [47] 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 [42] 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 tools have been developed within the OPEN/CAESAR environment, among which:
  - BISIMULATOR, which checks bisimulation equivalences and preorders,
  - CUNCTATOR, which performs on-the-fly 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 regular alternation-free μ-calculus 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_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 [50], CTL [37], ACTL [38], etc.) to be defined in XTL.

- PBG (*Partitioned BCG Graph*) is a file format implementing the theoretical concept of *Partitioned LTS* [46] 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 [55] 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 the VERTECS project team at Inria Rennes – Bretagne-Atlantique.

The CADP tools are well-integrated and can be accessed easily using either the EUCALYPTUS graphical interface or the SVL [43] 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.

5.2. The TRAIAN Compiler

Participants: Hubert Garavel [correspondent], Frédéric Lang.

We develop a compiler named TRAIAN 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 [44], being used in all recent compilers developed by the CONVECS team.

The TRAIAN compiler can be freely downloaded from the CONVECS Web site (see http://convecs.inria.fr/software/traian).

5.3. The PIC2LNT Translator

Participants: Radu Mateescu, Gwen Salaün [correspondent].

We develop a translator named PIC2LNT from an applied π-calculus (see Section 6.1) to LNT, which enables the analysis of concurrent value-passing mobile systems using CADP.

PIC2LNT is developed by using the SYNTAX tool (developed at Inria Paris-Rocquencourt) for lexical and syntactic analysis together with LOTOS NT for semantical aspects, in particular the definition, construction, and traversal of abstract trees.

The PIC2LNT translator can be freely downloaded from the CONVECS Web site (see http://convecs.inria.fr/software/pic2lnt).

6. New Results

6.1. New Formal Languages and their Concurrent Implementations

LNT is a next generation formal description language for asynchronous concurrent systems, which attempts to combine the best features of imperative programming languages and value-passing process algebras. LNT is increasingly used by the CONVECS team for industrial case studies and applications (see § 6.5) and serves also in university courses on concurrency, in particular at ENSIMAG (Grenoble) and at the Saarland University.

6.1.1. Translation from LNT to LOTOS

Participants: Hubert Garavel, Frédéric Lang, Wendelin Serwe.

The LNT2LOTOS, LNT.OPEN, and LPP tools convert LNT code to LOTOS, thus allowing the use of CADP to verify LNT descriptions. These tools have been used successfully for many different systems (see § 6.5 and § 9.1).

In 2012, in addition to 12 bug fixes, the following enhancements have been brought to these tools:

- We improved the ergonomy of the LNT2LOTOS translator by refining certain command-line options and by making some warning messages more user-friendly.
- We optimized the generated LOTOS code of the “disrupt” and “parallel” composition operators, so as to reduce the number of spurious warnings about impossible synchronizations and, more importantly, to meet the subset of LOTOS supported by the CAESAR compiler (static bound on the number of parallel processes).
- We improved the support for LNT programs that contain several modules by allowing the main process to be defined in any module (not only in the main module).
- We added new predefined functions for the generic data types (lists, sorted lists, and sets), and we updated accordingly the reference manual of LNT. The set types are now implemented correctly by avoiding duplicate elements.

6.1.2. Distributed Code Generation for Process Algebras

Participants: Hugues Evrard, Frédéric Lang.

One goal of CONVECS is to build a tool that generates automatically a distributed implementation of a system specified in LNT. This requires a protocol to realize process synchronization. As far as possible, this protocol must itself be distributed, so as to avoid the bottleneck that would inevitably arise if a unique process would have to manage all synchronizations in the system. A particularity of such a protocol is its ability to support *branching synchronizations*, corresponding to situations where a process may offer a choice of synchronizing actions (which themselves may nondeterministically involve several sets of synchronizing processes) instead of a single one. Therefore, a classical barrier protocol is not sufficient and a more elaborate synchronization protocol is needed.

In 2012, we explored the bibliography on synchronization protocols. Among almost twenty references studied, we selected three existing distributed synchronization protocols that seemed appropriate to our problem. In order to validate these protocols, we designed a tool chain that, given a system described as a parallel composition of LNT processes, generates an LNT specification of an implementation of the system (called the *implementation model*), by incorporating the protocol in the specification to realize the synchronizations. We then used CADP to check for livelocks and deadlocks possibly introduced in the implementation model by the protocol (using MCL and EVALUATOR 4.0), and to verify that the implementation model mimicks the behaviour of the system by equivalence checking (using BISIMULATOR).

Among the three protocols considered, we selected the most promising one [57], which is suitable for generalization to implement synchronization vectors (and hence, the generalized parallel composition operator of LNT). Using the methodology mentioned above, we discovered a previously unknown error in this protocol, which leads to deadlocks in certain situations, and we proposed a correction. An article has been submitted to an international conference.

6.1.3. Translation from an Applied Pi-Calculus to LNT

Participants: Radu Mateescu, Gwen Salaün.

The π-calculus is a process algebra defined by Milner, Parrow, and Walker two decades ago for describing concurrent mobile processes. So far, only a few verification tools have been designed for analyzing π-calculus specifications automatically. Our objective is to provide analysis features for π-calculus specifications by reusing the verification technology already available for value-passing process algebras without mobility. Our approach is based on a novel translation from the finite control fragment of π-calculus to LNT. To the best of our knowledge, this is the first π-calculus translation having a standard process algebra as target language.

In this work, we have also extended the original polyadic π-calculus with data-handling features. This results in a general-purpose applied π-calculus, which offers a good level of expressiveness for specifying mobile concurrent systems, and therefore for widening its possible application domains. As language for describing data types and functions, a natural choice was LNT itself: in this way, the data types and functions used in the π-calculus specification can be directly imported into the LNT code produced by translation.

The translation is fully automated by the tool PIC2LNT 2.0. This enables the analysis of applied π-calculus specifications using all verification tools of CADP, in particular the EVALUATOR 4.0 on-the-fly model checker, which evaluates temporal properties involving channel names and data values. PIC2LNT 2.0 was used for teaching mobile concurrency at Saarland University. A paper describing this work was accepted for publication in an international conference [16].

6.1.4. Translation from EB3 to LNT

Participants: Frédéric Lang, Radu Mateescu.

In collaboration with Dimitris Vekris (University Paris-Est Créteil), we have considered a translation from the EB3 language [39] for information systems to LNT. EB3 is inspired from a process algebra, but has the particularity to contain so-called *attribute functions*, whose semantics depend on the history of events. Therefore, the history of events becomes part of the state of an EB3 specification, which is unusual in process algebras.

Since EB3 is not equipped with native verification tools, we have proposed a translation from EB3 to LNT, which would enable EB3 specifications to be formally verified using CADP. Our formal translation scheme ensures the strong equivalence between the LTS corresponding to an EB3 specification and the LTS corresponding to the LNT code generated. The history of events is encoded as a particular LNT process "*memory*" synchronized on all EB3 events with the rest of the system. The memory process thus acts as a monitor that changes its state according to the occurring events and answers requests emitted by the attribute functions when needed. A prototype translator has been developed at University Paris-Est Créteil and a paper describing this work has been submitted to an international conference.

6.1.5. Coverage Analysis for LNT

Participants: Gwen Salaün, Lina Ye.

In the classic verification setting, we have an LNT specification of a system, a set of temporal properties to be verified on the LTS model corresponding to the LNT specification, and a data set of examples (test cases) we use for validation purposes. At this stage, building the set of validation examples and debugging the specification is a complicated task, in particular for non-experts.

Coverage analysis aims at proposing and developing techniques for automatically detecting parts of an LNT specification not (yet) covered during verification. Such LNT coverage analysis techniques would be very helpful for (i) extending the set of test cases with new inputs covering parts of the LNT specification that have not been analyzed yet, (ii) eliminating dead code in the LNT specification, and (iii) extending the set of temporal properties with new ones.

We have already identified four criteria (action, decision, block, property) and developed a prototype tool that automatically returns coverage values for these four criteria. We have applied our tool to LNT specifications of existing protocols, such as a reconfiguration protocol for component-based architectures [34], and found several cases of dead code and missing test cases.

6.1.6. Other Software Developments

Participants: Hubert Garavel, Frédéric Lang, Wendelin Serwe.

In addition of correcting 23 bugs in various CADP tools, we also brought the following enhancements:

- The EUCALYPTUS interface was improved regarding ergonomy and customization.
- The CADP tools for 32-bit and 64-bit Intel/Linux architectures were upgraded to use recent compilers and libraries, and CADP was modified to support Mac OS X 10.8 "Mountain Lion".
- The usability of the libraries for writing BCG files was improved to detect and signal an improper ordering of the primitives in application programs.
- The SYNTAX parser generator was improved by correcting two subtle errors, one of them causing an infinite looping on certain erroneous input programs. The CADP compilers developed using SYNTAX were enhanced to perform a better diagnosis of the situations when SYNTAX corrected syntactic errors automatically in erroneous programs.
- We improved an optimization of the CAESAR compiler for LOTOS, leading to a significant reduction of the execution time (from one hour and 51 minutes down to 58 seconds) for some examples of LOTOS programs with many variables. We optimized the CAESAR.OPEN script to invoke the CAESAR compiler directly whenever possible (instead of the GENERATOR tool), which improves the performance of graph generation, in particular for LNT.OPEN.
- Four demonstration examples of CADP were extended with LNT descriptions to illustrate the usage of the LNT language and of its compiler. Two examples have been simplified using the latest features of SVL, which can now handle the "n among m" parallel composition operator of LNT. Also, three examples have been reorganized for a better clarity and two couples of examples, which were closely related, have been merged into single examples.

6.2. Parallel and Distributed Verification

Participants: Hubert Garavel, Radu Mateescu, Wendelin Serwe.

For distributed verification, CADP provides the PBG format, which implements the theoretical concept of *Partitioned LTS* introduced in [46] and provides a unified access to an LTS distributed over a set of remote machines. The PBG format is equipped with the DISTRIBUTOR and PBG_MERGE (previously called BCG_MERGE [45]) tools, which perform the distributed generation of a partitioned LTS and the conversion of a partitioned LTS represented in the PBG format into a monolithic LTS stored in a BCG file.

To facilitate the manipulation of partitioned LTSs, CADP provides the PBG_CP, PBG_MV, and PBG_RM tools for copying, moving, and removing PBG files, maintaining consistency during these operations. The PBG_INFO tool provides several functionalities to inspect PBG files, such as checking consistency (i.e., existence and readability of all fragment files), calculating the size (number of states and transitions) of the corresponding LTS, displaying the list of labels, and concatenating remote log files (this is useful, e.g., to understand the reason why a PBG generation fails, and to compute global statistics about CPU and memory usage by the worker processes).

In 2012, in addition to correcting two bugs in DISTRIBUTOR and several bugs in the CAESAR_NETWORK_1 communication library used by the distributed verification tools, we also improved these tools as follows:

- We enhanced DISTRIBUTOR to support more than 256 distributed processes.
- We enhanced CAESAR_NETWORK_1 with a debugging facility, which enables traces of all distributed processes to be generated.
- We enhanced the graphical monitor of DISTRIBUTOR with the option of sorting the labels alphabetically, which facilitates their visual inspection.
- We extended PBG_INFO to enable the display of all labels in a partitioned LTS.

We also developed a prototype tool, named PBG_OPEN, which is an OPEN/CAESAR-compliant compiler for the PBG format, enabling the use of all CADP on-the-fly verification tools on a partitioned LTS. The main advantage of PBG_OPEN is that it can use the memory of several machines to store the transition relation of a partitioned LTS. Therefore, PBG_OPEN can explore on-the-fly large partitioned LTSs that could not be explored using other tool combinations. To reduce the amount of communications, PBG_OPEN can use a cache to store already encountered states, together with their outgoing transitions.

We experimented all these tools on the Grid'5000 computing infrastructure [35] using up to 512 distributed processes. These experiments confirmed the good scalability of our distributed LTS manipulation approach. A paper describing this work has been published in an international conference [12].

6.3. Timed, Probabilistic, and Stochastic Extensions

Participant: Hubert Garavel.

Process calculi provide a suitable formal framework for describing and analyzing concurrent systems, but need to be extended to model refined aspects of these systems. For instance, it may be necessary to represent probabilistic choices (in addition to deterministic and nondeterministic choices) as well as delays and latencies governed by probability laws. Many such extensions have been proposed in the literature, some of which have been implemented in software tools and applied to nontrivial problems. In particular, two of these extensions (namely, *Interactive Markov Chains* and *Interactive Probabilistic Chains*) are implemented in CADP. Despite these achievements, the state of the art is not satisfactory as the extended languages primarily focus on the probabilistic and stochastic aspects, leaving away the expressive and user-friendly features that process calculi provide for describing conventional concurrent systems.

In 2012, we undertook a study to merge probabilistic and stochastic aspects into modern high-level languages such as LNT. This work is done at Saarland University under the aegis of the Alexander von Humboldt foundation, in collaboration also with RWTH Aachen and Oxford University. We investigated the theoretical concepts, as well as their integration into modeling languages, together with the corresponding behavioural equivalences and temporal logics.

We also started experimenting with state-of-the-art software implementations, such as MODEST and PASS (Saarland University), COMPASS and MRMC (RWTH Aachen), and PRISM (Oxford University). Two of these tools (namely, MODEST and PRISM) have been used for lab exercises in the *Applied Concurrency Theory* block course created by H. Garavel at Saarland University. Following these experiments, evaluation reports have been produced, which provide feedback about issues and suggestions for enhancements. These reports have been addressed to the respective authors of each tool and already led to improvements in certain tools.

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

6.4.1. Compositional Model Checking

Participants: Frédéric Lang, Radu Mateescu.

We have continued our work on *partial model checking* following the approach proposed in [29]. Given a temporal logic formula φ to be evaluated on a set S of concurrent processes, partial model checking consists in transforming φ into another equivalent formula φ' to be evaluated on a subset of S. Formula φ' is constructed incrementally by choosing one process P in S and incorporating into φ the behavioral information corresponding to P – an operation called *quotienting*. Simplifications must be applied at each step, so as to maintain formulas at a tractable size.

In 2012, we have continued the development of our prototype tools for partial model checking of the regular alternation-free μ-calculus supporting all features of the input language of EXP.OPEN 2.1. We have also extended our work to handle useful fairness operators of alternation depth 2 in linear time, without developing the complex machinery that would be necessary to evaluate general μ-calculus formulas of alternation depth 2. A paper has been published in an international conference [15] and an extended version has been submitted to an international journal.

6.4.2. On-the-Fly Test Generation

Participants: Radu Mateescu, Wendelin Serwe.

In the context of the collaboration with STMicroelectronics (see § 6.5.1 and § 7.1), we studied techniques for testing if a (hardware) implementation is conform to a formal model written in LNT. Our approach is inspired by the theory of conformance testing [59], as implemented for instance in TGV [51] and JTorX [33].

We developed two prototype tools supporting conformance testing. The first tool implements a dedicated OPEN/CAESAR-compliant compiler for the particular asymmetric synchronous product of the model and the test purpose. This tool is a generic component for on-the-fly graph manipulation, taking as input two graphs and producing as output the graph of the asymmetric synchronous product. The second tool generates the complete test graph, which can be used to extract concrete test cases or to drive the test of the implementation. This tool was built from (slightly extended) existing generic components for on-the-fly graph manipulation (τ-compression and τ-confluence reductions, determinization, resolution of Boolean equation systems). The main advantage of our approach compared to existing tools is the use of LNT for test purposes, which facilitates the manipulation of data values.

6.5. Real-Life Applications and Case Studies

6.5.1. ACE Cache Coherency Protocol

Participants: Hubert Garavel, Abderahman Kriouile, Radu Mateescu, Wendelin Serwe.

In the context of a CIFRE convention with STMicroelectronics (see § 7.1), we studied the system-level cache coherency, a major challenge faced in the current system-on-chip architectures. Because of their increasing complexity (mainly due to the significant number of computing units), the validation effort using current simulation-based validation techniques grows exponentially. As an alternative, we study formal verification.

In 2012, we focused on the ACE (*AXI Coherency Extensions*) cache coherency protocol, a system-level coherency protocol proposed by ARM [25]. In a first step, we developed a formal LNT model (about 2600 lines of LNT) of a system consisting of an ACE-compliant cache coherent interconnect, processors, and a main memory. The model is parametric and can be instantiated with different configurations (number of processors, number of cache lines, number of memory lines) and different sets of supported elementary ACE operations, including an abstract operation that represents any other ACE operation. Using the OCIS simulator, we were able to explore the behavior of the system interactively, which has been found helpful by STMicroelectronics engineers.

Currently, our formal model supports a representative subset of five elementary operations of the ACE protocol (MakeUnique, ReadOnce, ReadShared, ReadUnique, and WriteBack). For each of these operations, we have written a liveness property in MCL expressing that the operation is executed until its termination. Using parametric SVL scripts (about 250 lines) and the EVALUATOR 4.0 model checker, we verified these properties on the fly for up to three memory lines and two processors with two cache lines each. We also generated the corresponding LTS (up to 250 million states and one billion transitions).

We also started considering data integrity properties. This required to translate a state-based property (namely, the consistency between the values stored in memory and in the local caches of the processors) into our action-based setting. This enabled us to automatically exhibit a known error present in a previous version of the ARM specification of the ACE protocol (which was corrected in a subsequent version of the specification). Using the LNT model corresponding to the latest version of the ACE specification, we spotted several potential data integrity issues that we reported to STMicroelectronics, where they are currently under investigation.

6.5.2. Realizability of Choreographies

Participants: Alexandre Dumont, Matthias Güdemann, Gwen Salaün.

Choreographies allow business and service architects to specify, with a global perspective, the requirements of applications built over distributed and interacting software entities. In collaboration with Pascal Poizat (University Paris-Sud), we proposed new techniques for verifying BPMN 2.0 choreographies, and particularly the *realizability* property. Realizability ensures that peers obtained via projection from a choreography interact as prescribed in the choreography requirements. Our approach is formally grounded on a model transformation into the LNT process algebra and the use of equivalence checking. It is also completely tool-supported through interaction with the Eclipse BPMN2 modeler and CADP. These results have been published in an international conference [17].

In collaboration with Meriem Ouederni (University of Nantes), we extended our techniques for analyzing choreographies to restore realizability for non-realizable, but *repairable* choreographies. For this we exploit the counterexamples generated by the equivalence checker BISIMULATOR to identify problematic messages in the choreography. For those messages we add distributed local monitors to the system which delay message sending if necessary, to restore correct message sequences. This iterative approach introduces the minimal number of necessary additional messages to restore realizability, and the monitors are generated in the most permissive way, i.e., by considering all possible interleavings given the behaviour of the peers participating to the choreography. It is fully automated by a prototype tool we implemented. These results have been published in an international conference [14].

We developed a common formal description language, named CIF (*Choreography Intermediate Format*), for the verification of choreographies. CIF is based on an XML representation for easy exchange between programs, an XSD schema for validation, and a translation to LNT for verification. CIF is used as an intermediate language to specify choreographies, but can also serve as target language for translating various choreography specification languages, such as BPMN 2.0. The back-end connection to CADP via LNT enables the automation of some key choreography analysis tasks (repairability, realizability, conformance, etc.). Our framework is extensible with other front-end and back-end connections to, respectively, other choreography languages and formal verification tools.

6.5.3. Self-Configuration Protocol for Distributed Cloud Applications

Participants: Rim Abid, Gwen Salaün.

We collaborate with Noël de Palma and Fabienne Boyer (University Joseph Fourier), Xavier Etchevers and Thierry Coupaye (Orange Labs) in the field of cloud computing applications, which are complex, distributed artifacts involving multiple software components running on separate virtual machines. Setting up, (re)configuring, and monitoring these applications are complicated tasks because a software application may depend on several remote software and virtual machine configurations. These management tasks involve many complex protocols, which fully automate these tasks while preserving application consistency as well as some key properties.

In this work, we focus on a self-configuration protocol, which is able to configure a whole distributed application without requiring any centralized server. The high degree of parallelism involved in this protocol makes its design complicated and error-prone. In order to check that this protocol works as expected, we specify it in LNT and verify it using the CADP toolbox. The use of these formal techniques and tools helped to detect a bug in the protocol, and served as a workbench to experiment with several possible communication models. These results led to a publication in an international conference [18].

We are currently studying two variants of the self-configuration protocol, one handling virtual machine failures, and one allowing dynamicity in the system (addition and removal of virtual machines) using a publish-subscribe communication framework.

6.5.4. Networks of Programmable Logic Controllers

Participants: Hubert Garavel, Fatma Jebali, Jingyan Jourdan-Lu, Frédéric Lang, Eric Léo, Radu Mateescu.

In the context of the Bluesky project (see § 8.1.2.1), we study the software applications embedded on the PLCs (*Programmable Logic Controllers*) manufactured by Crouzet Automatismes. One of the objectives of Bluesky is to enable the rigorous design of complex control applications running on several PLCs connected by a network. Such applications are instances of GALS (*Globally Asynchronous, Locally Synchronous*) systems composed of several synchronous automata embedded on individual PLCs, which interact asynchronously by exchanging messages. A formal analysis of these systems can be naturally achieved by using the formal languages and verification techniques developed in the field of asynchronous concurrency.

For describing the applications embedded on individual PLCs, Crouzet provides a dataflow language with graphical syntax and synchronous semantics, equipped with an ergonomic user interface that facilitates the learning and use of the language by non-experts. To equip the PLC language of Crouzet with functionalities for automated verification, the solution adopted in Bluesky was to translate it into a pivot language (to be defined within the project) that will enable the connection to testing and verification tools covering the synchronous and asynchronous aspects. Our work focuses on the translation from the pivot language to LNT, which will provide a direct connection to all verification functionalities of CADP, namely model checking and equivalence checking.

In 2012, in interaction with Crouzet engineers, we studied the PLC language of Crouzet to understand precisely its static and dynamic semantics. We specified manually in LNT several examples of control applications provided by Crouzet, with the goal of identifying the principles of translating the PLC language of Crouzet to LNT. We formulated in MCL several safety and liveness properties concerning the temporal ordering of input and output events by the control applications, and we successfully verified them on the LNT specifications. We also started to study the network communication mechanisms between PLCs to identify a suitable LNT abstraction of the communication layer.

6.5.5. Other Case Studies

Participants: Frédéric Lang, Radu Mateescu, Wendelin Serwe.

Continuing a study [53] started in the context of the Multival project (see http://vasy.inria.fr/multival), we considered the Platform 2012 architecture proposed by STMicroelectronics, focusing on the Dynamic Task Dispatcher (DTD), a hardware block that assigns a set of application tasks to a set of processors. In 2012, we extended our LNT model and the corresponding MCL properties in order to handle heterogeneous processors equipped with different kinds of processor extensions. We also used constraints on the initialization phase, which reduced the size of the LTS by a factor of up to ten and hence enabled the generation of the LTS for

up to eight processors (instead of only six). Both extensions together enabled to discover the possibility of a livelock.

We attempted to investigate this issue further by cosimulation (using the EXEC/CAESAR framework) with the original C++ model of the architect. Unfortunately, the C++ model did not behave correctly for the particular aforementioned application scenario. It was not possible to change this model because the recent evolutions of Platform 2012 excluded the DTD, as its requirements in terms of silicon surface were considered too large. This work, including the LNT model as appendix, has been accepted for publication in an international journal [5].

In collaboration with Nuno Mendes and Claudine Chaouiya (Gulbenkian Institute, Portugal), Yves-Stan Le Cornec (IBISC, University Evry Val d'Essonne) and Grégory Batt (CONTRAINTES project-team, Inria Paris-Rocquencourt), we have studied the use of CADP for checking the reachability of stable states in genetic regulatory networks. A compositional and logical model of genetic regulatory networks called *logical regulatory modules* was defined and translated to LNT processes and EXP.OPEN 2.1 networks of LTSs. Compositional minimization modulo safety equivalence was applied to the generated network, so as to palliate state explosion while preserving the reachability property. The approach has been illustrated on the segment polarity module involved in the segmentation of the fruit fly embryo and on the delta-notch module involved in cell differentiation in crucial steps of embryonic development of several species. A paper has been submitted to an international journal.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Grants with Industry

Participants: Hubert Garavel, Abderahman Kriouile, Radu Mateescu, Wendelin Serwe.

Abderahman Kriouile is supported by a CIFRE PhD grant (from March 2012 to March 2015) from STMicroelectronics (Grenoble) on the verification of cache coherency in systems on chip, under the supervision of Grégory Faux and Massimo Zendri (STMicroelectronics), Radu Mateescu and Wendelin Serwe (CONVECS).

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. FSN (Fonds National pour la Société Numérique)

8.1.1.1. OpenCloudware

Participants: Rim Abid, Hugues Evrard, Frédéric Lang, Gwen Salaün [correspondent], Lina Ye.

OpenCloudware (see http://www.opencloudware.org) is a project funded by the FSN. The project is led by France Telecom / Orange Labs (Meylan, France) and involves 18 partners, among which Bull, OW2, Thalès, Inria, etc. OpenCloudware aims at providing an open software platform enabling the development, deployment and administration of cloud applications. The objective is to provide a set of integrated software components for (i) modeling distributed applications to be executed on cloud computing infrastructures, (ii) developing and constructing multi-tier virtualized applications, and (iii) deploying and administrating these applications (PaaS platform) possibly on multi-IaaS infrastructures.

OpenCloudware started in January 2012 for three years and nine months. The main contributions of CONVECS to OpenCloudware are the formal specification of the models, architectures, and protocols (self-deployment, self-management, etc.) underlying the OpenCloudware platform, the automated generation of code from these specifications for rapid prototyping purposes, and the formal verification of the aforementioned protocols.

8.1.1.2. Connexion

Participants: Hubert Garavel [correspondent], Frédéric Lang, Raquel Oliveira.

Connexion (*COntrôle commande Nucléaire Numérique pour l'EXport et la rénovatION*) is a project funded by the FSN within the second call for projects "*Investissements d'Avenir — Briques génériques du logiciel embarqué*". The project (see http://www.cluster-connexion.fr), led by EDF and supported by the *Pôles de compétitivité* Minalogic, Systematic, and *Pôle Nucléaire Bourgogne*, involves many industrial and academic partners, namely All4Tech, Alstom Power, ArevA, Atos Worldgrid, CEA, CNRS/CRAN, Corys Tess, ENS Cachan, Esterel Technologies, Inria, LIG, Predict, and Rolls-Royce. Connexion aims at proposing and validating an innovative architecture dedicated to the design and implementation of control systems for new nuclear power plants in France and abroad.

Connexion started in April 2012 for four years. CONVECS will participate, in cooperation with the IIHM team of LIG, to study the application of CADP to specify and validate human-machine interfaces formally.

8.1.2. Competitivity Clusters

8.1.2.1. Bluesky for I-Automation

Participants: Hubert Garavel, Fatma Jebali, Jingyan Jourdan-Lu, Frédéric Lang, Eric Léo, Radu Mateescu [correspondent].

Bluesky for I-Automation is a project funded by the FUI (*Fonds Unique Interministériel*) within the *Pôle de Compétitivité* Minalogic. The project, led by Crouzet Automatismes (Valence), involves the SMEs (*Small and Medium Enterprises*) Mootwin and VerticalM2M, the LCIS laboratory of Grenoble INP, and CONVECS. Bluesky aims at bringing closer the design of automation applications and the Internet of things by providing an integrated solution consisting of hardware, software, and services enabling a distributed, Internet-based design and development of automation systems. The automation systems targeted by the project are networks of programmable logic controllers, which belong to the class of GALS (*Globally Asynchronous, Locally Synchronous*) systems.

Bluesky started in September 2012 for three years. The main contributions of CONVECS to Bluesky are the definition of the formal pivot language for describing the asynchronous behaviour of logic controller networks and the automated verification of the behaviour using compositional model checking and equivalence checking techniques.

8.1.3. Other National Collaborations

Additionally, we collaborated in 2012 with the following Inria project-teams:

- CONTRAINTES (Inria Paris-Rocquencourt): Grégory Batt,
- OASIS (Inria Sophia-Antipolis – Méditerranée): Eric Madelaine and Ludovic Henrio.

Beyond Inria, we had sustained scientific relations with the following researchers:

- Gaëlle Calvary and Sophie Dupuy-Chessa (LIG, Grenoble),
- Pascal Poizat (LIP6, Paris),
- Meriem Ouederni (IRIT, Toulouse),
- Dimitris Vekris (LACL, Paris-Est Créteil).

8.2. European Initiatives

8.2.1. FP7 Projects

8.2.1.1. Sensation

Participants: Hubert Garavel, Radu Mateescu, Wendelin Serwe.

Sensation (*Self ENergy-Supporting Autonomous computaTION*) is the European project no. 318490 funded by the FP7-ICT-11-8 programme. The project (see http://people.cs.aau.dk/~rrh/SENSATION) gathers 9 participants: Inria (Triskell and Convecs teams), Aalborg University (Denmark), RWTH Aachen and Saarland University (Germany), University of Twente and Embedded System Institute (The Netherlands), STMicroelectronics (France), GomSpace (Denmark), and Recore Systems (The Netherlands). The main goal of Sensation is to increase the scale of systems that are self-supporting by balancing energy harvesting and consumption up to the level of complete products. In order to build such Energy Centric Systems, embedded system designers face the quest for optimal performance within acceptable reliability and tight energy bounds. Programming systems that reconfigure themselves in view of changing tasks, resources, errors and available energy is a demanding challenge.

Sensation started on October 1st, 2012 for three years. CONVECS contributes to the project regarding the extension of formal languages with quantitative aspects, studying common semantic models for quantitative analysis, and applying formal modeling and analysis to the case studies provided by the industrial partners.

8.2.2. Collaborations with Major European Organizations

The CONVECS team is member of the FMICS (*Formal Methods for Industrial Critical Systems*) working group of ERCIM (see http://fmics.inria.fr). R. Mateescu is currently the chairman of the FMICS working group and H. Garavel is member of the FMICS board, in charge of dissemination actions.

Hubert Garavel was appointed to a new Working Group within Informatics Europe: "Parallel Computing (Supercomputing) Education in Europe: State-of-Art". This is a relatively small working group (about 10 people) with the following missions: to show the need for urgent changes in higher education in the area of computational sciences, to compose a survey of the current landscape of parallel computing and supercomputing education in Europe with respect to different universities and countries, and to prepare a set of recommendations on how to bring ideas of parallel computing and supercomputing into higher educational systems of European countries.

8.2.3. Other European Collaborations

In addition to our partners in aforementioned contractual collaborations, in 2012 we had scientific relations with several European universities and research centers, including:

- Saarland University (Alexander Graf-Brill, Ernst-Moritz Hahn, Arnd Hartmanns, Holger Hermanns, and Andrea Turrini),
- Oxford University (Ernst-Moritz Hahn, Marta Kwiatkowska, and Dave Parker),
- RWTH Aachen (Joost-Pieter Katoen and Viet Yen Nguyen),
- University of Twente (Freark van der Berg and Marielle Stoelinga),
- Technical University of Eindhoven (Anton Wijs).

H. Garavel participates in the DFG (*Deutsche Forschungsgemeinshaft*) transregional project AVACS (*Automatic Verification and Analysis of Complex Systems*, see http://www.avacs.org) and he attended two meetings held at Freiburg (Germany) in February 2012 and at Mannheim (Germany) in November 2012.

8.3. International Initiatives

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

8.3.1. Other International Collaborations

We had sustained scientific relations with Tevfik Bultan (University of California at Santa Barbara, USA).

8.4. International Research Visitors

8.4.1. Visits of International Scientists

- Pascal Poizat (LIP6, University Pierre et Marie Curie, Paris) visited us on March 26–27, 2012.
- Dimitris Vekris (LACL, University Paris-Est Créteil) visited us from April 23 to May 11, 2012.
- Meriem Ouederni (IRIT, Toulouse) visited us on June 4–15, 2012.
- The annual CONVECS seminar was held in Pont-en-Royans (France) on November 5–7, 2012. The following invited scientists attended the seminar:
  - Jérémy Buisson (University of Bretagne-Sud / VALORIA and Ecoles de St-Cyr Coëtquidan) gave on November 5, 2012 a talk entitled "*Vers un futur pi-ADL reconfigurable*".
  - Sophie Dupuy-Chessa (LIG, Grenoble) gave on November 6, 2012 a talk entitled "*Qualité des interfaces homme-machine plastiques*".
  - Massimo Zendri (STMicroelectronics) gave on November 6, 2012 a talk entitled "*Circuit Level Formal Verification in Industrial Environment*".

9. Dissemination

9.1. Scientific Animation

9.1.1. Software Dissemination and Internet Visibility

The CONVECS team distributes several software tools: the CADP toolbox (see 5.1), the TRAIAN compiler (see 5.2), and the PIC2LNT translator (see 5.3). In 2012, the main facts are the following:

- We prepared and distributed 11 successive beta-versions (from 2010-h to 2010-m, from 2011-a to 2011-c, and from 2012-a to 2012-b "Zurich") of CADP.
- We were requested to grant CADP licenses for 1227 different computers in the world.
- We released version 2.7 of the TRAIAN compiler for LOTOS NT in November 2012.
- We released version 2.0 of the PIC2LNT translator from an applied π-calculus to LNT in October 2012.

We constructed the CONVECS Web site (see http://convecs.inria.fr) and we updated it with scientific contents, announcements, publications, etc.

By the end of December 2012, the CADP forum (see http://cadp.inria.fr/forum.html), opened in 2007 for discussions regarding the CADP toolbox, had over 230 registered users and over 1400 messages had been exchanged.

Other research teams took advantage of the software components provided by CADP (e.g., the BCG and OPEN/CAESAR environments) to build their own research software. We can mention the following developments:

- Formal modeling and verification of BPEL-based Web service composition [62],
- A specific-domain design tool for FPGA-based image and video processing system [63],
- Quantitative timed analysis of interactive Markov chains [49],
- Synchronizability for verification of asynchronously communicating systems [31],
- Deciding choreography realizability [30],
- Automata learning through counterexample guided abstraction refinement [26],
- The Reo+mCRL2 framework for model-checking dataflow in service compositions [52],
- The IDCM analysis tool for components and architectures dedicated to incremental construction [54],
- Rigorous development of composite grid services [60],
- Integrating model-based testing and analysis tools via test case exchange [28], and
- The DFTCalc tool for calculating the failure probability of a dynamic fault tree (DFT) using LNT [61].

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

- Learning and testing the bounded retransmission protocol [27],
- Specification and validation of a real-time simple parallel kernel for dependable distributed systems [40],
- Abstraction-based malware analysis using rewriting and model checking [32], and
- Model checking of scenario-aware dataflow [58].

9.1.2. Program Committees

In 2012, the members of CONVECS took on the following responsibilities:

- H. Garavel is an editorial board member of STTT (Springer International Journal on Software Tools for Technology Transfer).
- F. Lang was a program committee member for NEPTUNE'2012 (Nice Environment with a Process and Tools Using Norms and Example), Paris, France, June 6-7, 2012.
- F. Lang was a program committee member for ESOCC'2012 (European Conference on Service-Oriented and Cloud Computing), Bertinoro, Italy, September 19-21, 2012.
- F. Lang and G. Salaün were program committee members for FMICS'2012 (17h International Workshop on Formal Methods for Industrial Critical Systems), Paris, France, August 27-28, 2012.
- R. Mateescu was a program committee member for GRAPHITE'2012 (1st International Workshop on Graph Inspection and Traversal Engineering), Talinn, Estonia, March 31 - April 1st, 2012.
- R. Mateescu was a program committee member for TASE'2012 (6th International Symposium on Theoretical Aspects of Software Engineering), Beijing, China, July 4-6, 2012.
- G. Salaün is an editorial board member of SOCA (Springer International Journal on Service Oriented Computing and Applications).
- G. Salaün was a program committee member for COORDINATION'2012 (14th International Conference on Coordination Models and Languages), Stockholm, Sweden, June 14-15, 2012.
- G. Salaün was a program committee chair for FACS'2012 (9th International Symposium on Formal Aspects of Component Software), Mountain View, USA, September 12-14, 2012.
- G. Salaün was a program committee member for FLACOS'2012 (6th International Workshop on Formal Languages and Analysis of Contract-Oriented Software), Bertinoro, Italy, September 19, 2012.
- G. Salaün was a steering committee member for FOCLASA'2012 (11th International Workshop on Foundations of Coordination Languages and Self-adaptation), Newcastle upon Tyne, UK, September 8, 2012.
- W. Serwe was a program committee member for PDMC'2012 (11th International Workshop on Parallel and Distributed Methods in Verification), Imperial College, London, UK, September 17, 2012.

9.1.3. Awards and Distinctions

- H. Garavel is an invited professor at the University of Saarland (Germany) after he received the Gay-Lussac Humboldt Prize in 2011.
- M. Güdemann received the Software Engineering Award of the Ernst Denert Association in 2012.

9.1.4. Lectures and Invited Conferences

- H. Garavel gave a keynote lecture entitled "*Three Decades of Success Stories in Formal Methods*" at FMICS'2012 (Paris, France) on August 28, 2012.
- H. Garavel gave a lecture entitled "*Trois décennies de réussite en méthodes formelles*" in the Aerospace Valley industrial conference on formal methods (Toulouse, France) on November 13, 2012.

- M. Güdemann visited the Institute for Systems and Systems Engineering (ISSE) at the University of Augsburg (Germany) on March 1st, 2012. He gave a tutorial on CADP.
- F. Lang and W. Serwe presented a tutorial on CADP at AFADL'2012 (Grenoble, France) on January 12, 2012.
- R. Mateescu visited the LRI laboratory (University Paris-Sud at Evry) on March 21–22, 2012. He gave a talk entitled "*Extending Temporal Logics in Practice: Model Checking and Applications*" on March 22, 2012.
- R. Mateescu presented a tutorial on CADP at FM'2012 (Paris, France) on August 28, 2012.
- R. Mateescu gave a talk entitled "*Querying Graphs On-the-Fly using a Model Checking Language*" at the LIG laboratory (Grenoble) on September 27, 2012.
- G. Salaün visited the LRI laboratory (University Paris-Sud at Evry) on March 19, 2012. He gave a talk entitled "*Specifying and Verifying a Self-Configuration Protocol for Distributed Applications in the Cloud using LNT and CADP*" on March 19, 2012.
- G. Salaün visited the LINA laboratory (University of Nantes) on March 22–23, 2012. He gave a talk entitled "*Specifying and Verifying a Self-configuration Protocol for Distributed Applications in the Cloud using LNT and CADP*" on March 22, 2012.
- G. Salaün visited the FBK laboratory (Trento, Italy) on March 28, 2012. He gave a talk entitled "*Design and Verification of Distributed Systems*" on March 28, 2012.
- G. Salaün visited the IRIT laboratory (Toulouse) on November 19–21, 2012. He gave a talk entitled "*Design and Verification of Communicating Systems*" on November 19, 2012.
- W. Serwe participated to the kick-off meeting of the Sensation project, held on November 1–2, 2012 in Aalborg (Denmark). He gave a talk entitled "*Model Checking and Performance Evaluation with CADP Illustrated on Shared-Memory Mutual Exclusion Protocols*" on November 2, 2012.
- W. Serwe participated to the Grid'5000 winter school, held on December 3–6, 2012 in Nantes. He gave a talk entitled "*Large Scale Distributed Verification using CADP on Grid'5000*" on December 6, 2012.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

CONVECS is a host team for the computer science master entitled "*Mathématiques, Informatique, spécialité : Systèmes et Logiciels*", common to Grenoble INP and University Joseph Fourier.

In 2012, we carried out the following teaching activities:

- H. Evrard served as a teaching assistant in a course on "*Algorithmique et structures de données*", given by Frédéric Wagner to the first year computer science engineering students of ENSIMAG (18 hours).
- H. Evrard served as a teaching assistant in a course on "*Introduction aux réseaux de communication*", given by Roland Groz to the first year computer science engineering students of ENSIMAG (18 hours).
- H. Evrard served as a teaching assistant in a course on "*Systèmes d'exploitation et programmation concurrente*", given by Yves Denneulin to the first year computer science engineering students of ENSIMAG (18 hours).
- H. Garavel created a block course entitled "*Advanced Concurrency Theory*" for the bachelor and master students of Saarland University (4 ECTS credits, assistant: Alexander Graf-Brill).
- A. Kriouile served as a teaching assistant in a course on "*Conception de circuits et architecture des ordinateurs*", given by Frédéric Pétrot to the first year computer science engineering students of ENSIMAG (18 hours).

- F. Lang and W. Serwe gave, jointly with Pascal Raymond (CNRS, Verimag), a course on "*Méthodes formelles de développement*" to the computer science engineering students of CNAM ("*Conservatoire National des Arts et Métiers*") Grenoble (27 hours).
- F. Lang and W. Serwe gave a course on "*Spécification et vérification de systèmes concurrents et temps-réel*" to the third year computer science engineering students of ENSIMAG (18 hours).
- G. Salaün gave lectures on "*Algorithmics and Object-Oriented Programming*" to the 2nd year computer science engineering students of ENSIMAG (36 hours).
- G. Salaün is co-responsible of the ISI (*Ingénierie des Systèmes d'Information*) department of ENSIMAG since September 1, 2011.

9.2.2. Juries

- G. Salaün was a panel member for Jose Antonio Martin Baena's PhD thesis, entitled "*Secure Adaptation of Software Services*", defended at the University of Málaga, Spain, on July 24, 2012.
- W. Serwe was a core committee / reading committee member of F. P. M. Stapper's PhD thesis, entitled "*Bridging Formal Methods: An Engineering Perspective*", defended at the Technical University of Eindhoven, The Netherlands, on November 8, 2012.

9.3. Popularization

H. Garavel participates to the committee in charge of organizing the Aerospace Valley series of industrial conferences on formal methods. The first conference, held on November 13, 2012, was a success, as it attracted 85 participants, among which 61 industrialists from 33 different companies.

R. Mateescu published a report about the 17th workshop on *Formal Methods for Industrial Critical Systems* [21].

In the context of the 20th anniversary of the Inria research center in Montbonnot, H. Garavel contributed to the collection of articles describing the evolution of computer sciences during the last 20 years [20].

9.4. Miscellaneous Activities

Within the Minalogic *Pôle de compétitivité mondial*, H. Garavel is a member of the operational committee of the EMSOC cluster (Embedded System on Chip).

H. Garavel was a reviewer for the German Excellence Initiative (Engineering Sciences Review Panel ING 14 for Clusters of Excellence), Bonn (Germany), January 10–12, 2012.

H. Garavel was a member of the working group in charge of restructuring the scientific themes of the LIG laboratory.

H. Garavel, F. Lang, R. Mateescu, G. Salaün, and W. Serwe attended the Inria evaluation seminar for teams working on "Embedded Systems and Real-Time", Rungis/Orly (France), March 19–20, 2012.

H. Garavel participated in the 40th Symposium for Research Award Winners of the Alexander von Humboldt Foundation, Bamberg (Germany), March 22–25, 2012.

H. Garavel participated in the "Cérémonie de remise du Prix Gay-Lussac Humboldt 2011", Académie des Sciences – Institut de France, Paris, April 10, 2012.

H. Garavel participated in the Annual Meeting (*Jahrestagung*) of the Alexander von Humboldt Foundation, Berlin, June 19–21, 2012.

H. Garavel was a reviewer for the "*Futur et Ruptures*" joint programme of Institut Mines-Télécom and Fondation Télécom.

F. Lang is a member of the "*commission du développement technologique*", which is in charge of selecting R&D projects for Inria Grenoble – Rhône-Alpes.

R. Mateescu is the correspondent of the "*Département des Partenariats Européens*" for Inria Grenoble – Rhône-Alpes.

G. Salaün was a member of the council of ENSIMAG until October 2012.

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

G. Salaün is a member of the scientific council of Grenoble INP (*Conseil scientifique de l'institut*).

10. Bibliography

Publications of the year

Articles in International Peer-Reviewed Journals

[1] C. CANAL, J. CÁMARA, G. SALAÜN. *Structural Reconfiguration of Systems under Behavioral Adaptation*, in "Science of Computer Programming", September 2012, vol. 78, n° 1, p. 46-64, http://hal.inria.fr/hal-00734057.

[2] J. CÁMARA, G. SALAÜN, C. CANAL, M. OUEDERNI. *Interactive specification and verification of behavioral adaptation contracts*, in "Information and Software Technology", July 2012, vol. 54, n° 7, p. 701-723, http://hal.inria.fr/hal-00694516.

[3] F. DURÁN, M. OUEDERNI, G. SALAÜN. *A generic framework for n-protocol compatibility checking*, in "Science of Computer Programming", July 2012, vol. 77, n° 7-8, p. 870-886, http://hal.inria.fr/hal-00694561.

[4] H. GARAVEL, F. LANG, R. MATEESCU, W. SERWE. *CADP 2011: A Toolbox for the Construction and Analysis of Distributed Processes*, in "International Journal on Software Tools for Technology Transfer", 2012 [*DOI :* 10.1007/s10009-012-0244-z], http://hal.inria.fr/hal-00715056.

[5] E. LANTREIBECQ, W. SERWE. *Formal Analysis of a Hardware Dynamic Task Dispatcher with CADP*, in "Science of Computer Programming", 2013, to appear.

[6] R. MATEESCU, P. POIZAT, G. SALAÜN. *Adaptation of Service Protocols using Process Algebra and On-the-Fly Reduction Techniques*, in "IEEE Transactions on Software Engineering", 2012 [*DOI :* 10.1109/TSE.2011.62], http://hal.inria.fr/hal-00717252.

[7] R. MATEESCU, W. SERWE. *Model Checking and Performance Evaluation with CADP Illustrated on Shared-Memory Mutual Exclusion Protocols*, in "Science of Computer Programming", February 2012 [*DOI :* 10.1016/j.scico.2012.01.003], http://hal.inria.fr/hal-00671321.

[8] R. MATEESCU, A. WIJS. *Sequential and distributed on-the-fly computation of weak tau-confluence*, in "Science of Computer Programming", 2012, vol. 77, n° 10-11, p. 1075-1094 [*DOI :* 10.1016/j.scico.2011.07.004], http://hal.inria.fr/hal-00676451.

[9] K. H. ROSE, R. BLOO, F. LANG. *On Explicit Substitution with Names*, in "Journal of Automated Reasoning", August 2012, vol. 49, n° 2, p. 275-300 [*DOI :* 10.1007/s10817-011-9222-5], http://hal.inria.fr/hal-00763399.

[10] G. SALAÜN, T. BULTAN, N. ROOHI. *Realizability of Choreographies using Process Algebra Encodings*, in "IEEE Transactions on Services Computing", August 2012, vol. 5, n^{o} 3, p. 290-304, http://hal.inria.fr/hal-00726448.

International Conferences with Proceedings

[11] S. ALATARTSEV, M. GÜDEMANN, F. ORTMEIER. *Trajectory Description Conception for Industrial Robots*, in "Proceedings of the 7th German Conference on Robotics (ROBOTIK 2012)", Munich, Germany, VDE Verlag, May 2012, p. 365-370, http://hal.inria.fr/hal-00727303.

[12] H. GARAVEL, R. MATEESCU, W. SERWE. *Large-Scale Distributed Verification using CADP: Beyond Clusters to Grids*, in "11th International Workshop on Parallel and Distributed Methods in verifiCation", London, United Kingdom, September 2012, http://hal.inria.fr/hal-00730668.

[13] M. GÜDEMANN, M. LIPACZEWSKI, S. STRUCK, F. ORTMEIER. *Unifying Probabilistic and Traditional Formal Model Based Analysis*, in "8. Dagstuhl-Workshop MBEES 2012 - Model-Based Development of Embedded Systems", Dagstuhl, Germany, February 2012, http://hal.inria.fr/hal-00665607.

[14] M. GÜDEMANN, G. SALAÜN, M. OUEDERNI. *Counterexample Guided Synthesis of Monitors for Realizability Enforcement*, in "Automated Technology for Verification and Analysis - 10th International Symposium, ATVA 2012", India, LNCS, Springer, October 2012, vol. 7561, p. 238-253 [*DOI :* 10.1007/978-3-642-33386-6_20], http://hal.inria.fr/hal-00742159.

[15] F. LANG, R. MATEESCU. *Partial Model Checking using Networks of Labelled Transition Systems and Boolean Equation Systems*, in "Tools and Algorithms for the Construction and Analysis of Systems TACAS'2012", Tallinn, Estonia, C. FLANAGAN, B. KÖNIG (editors), LNCS, Springer, March 2012, vol. 7214, p. 141-156, http://hal.inria.fr/hal-00684471.

[16] R. MATEESCU, G. SALAÜN. *PIC2LNT: Model Transformation for Model Checking an Applied Pi-Calculus*, in "Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems TACAS'2013 (Rome, Italy)", N. PITERMAN, S. SMOLKA (editors), LNCS, Springer, March 2013, to appear.

[17] P. POIZAT, G. SALAÜN. *Checking the Realizability of BPMN 2.0 Choreographies*, in "27th Symposium On Applied Computing (SAC 2012)", Italy, March 2012, p. 1927-1934, http://hal.inria.fr/hal-00685393.

[18] G. SALAÜN, X. ETCHEVERS, N. DE PALMA, F. BOYER, T. COUPAYE. *Verification of a Self-configuration Protocol for Distributed Applications in the Cloud*, in "27th Symposium On Applied Computing (SAC 2012)", Italy, March 2012, p. 1278-1283, http://hal.inria.fr/hal-00685394.

[19] S. STRUCK, M. LIPACZEWSKI, F. ORTMEIER, M. GÜDEMANN. *Multi-Objective Optimization of Formal Specifications*, in "Proceedings of the 14th High Assurance System Engineering Symposium (HASE 2012)", Omaha, United States, IEEE, October 2012, p. 201-208 [*DOI :* 10.1109/HASE.2012.21], http://hal.inria.fr/hal-00735640.

Scientific Popularization

[20] H. GARAVEL, I. BELLIN. *La fiabilité des systèmes devient un défi majeur*, November 2012, Article series for the 20th anniversary of the Inria research center in Montbonnot, http://www.inria.fr/centre/grenoble/actualites/la-fiabilite-des-systemes-devient-un-defi-majeur.

[21] R. MATEESCU. *17th International Workshop on Formal Methods for Industrial Critical Systems*, in "ERCIM News", October 2012, vol. 91, http://ercim-news.ercim.eu/en91/jea/17th-international-workshop-on-formal-methods-for-industrial-critical-systems.

Other Publications

[22] A. DUMONT. *A LOTOS NT Library for Modeling, Analysis, and Validation of Distributed Systems*, Ecole Nationale Supérieure d'Informatiques et Mathématiques Appliquées de Grenoble ENSIMAG, 2012.

[23] H. GARAVEL, F. LANG, R. MATEESCU, G. SALAÜN, W. SERWE. *CADP : une boîte à outils pour la conception et l'analyse de systèmes distribués*, January 2012, This tutorial was presented at 11èmes Journées Francophones sur les Approches Formelles dans l'Assistance au Développement de Logiciels AFADL'2012 (Grenoble, France), January 11-13, 2012., http://hal.inria.fr/hal-00667288.

[24] H. GARAVEL, F. LANG, R. MATEESCU, W. SERWE. *CADP Tutorial*, 2012, This tutorial was presented at the 18th International Symposium on Formal Methods FM'2012 (Paris, France), August 27-31, 2012., http://hal.inria.fr/hal-00764932.

References in notes

[25] *AMBA AXI and ACE Protocol Specification*, ARM IHI 0022D (ID102711), ARM, October 22 2011.

[26] F. AARTS, F. HEIDARIAN, H. KUPPENS, P. OLSEN, F. W. VAANDRAGER. *Automata Learning through Counterexample Guided Abstraction Refinement*, in "Proceedings of the 18th International Symposium on Formal Methods FM'2012 (Paris, France)", D. GIANNAKOPOULOU, D. MÉRY (editors), LNCS, Springer, August 2012, vol. 7436, p. 10–27.

[27] F. AARTS, H. KUPPENS, J. TRETMANS, F. W. VAANDRAGER, S. VERWER. *Learning and Testing the Bounded Retransmission Protocol*, in "Proceedings of the 11th International Conference on Grammatical Inference ICGI'2012 (College Park, Maryland, USA)", S. OSSOWSKI, P. LECCA (editors), JMLR Workshop and Conference Proceedings (Open Access), March 2012, vol. 21, p. 4–18.

[28] B. AICHERNIG, F. LORBER, S. TIRAN. *Integrating Model-Based Testing and Analysis Tools via Test Case Exchange*, in "Proceedings of the 6th International Symposium on Theoretical Aspects of Software Engineering TASE'2012 (Beijing, China)", T. MARGARIA, Z. QIU, H. YANG (editors), IEEE Press, July 2012, p. 119–126.

[29] H. R. ANDERSEN. *Partial Model Checking*, in "Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science LICS (San Diego, California, USA)", IEEE Computer Society Press, June 1995, p. 398–407.

[30] S. BASU, T. BULTAN, M. OUEDERNI. *Deciding Choreography Realizability*, in "Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages POPL'2012 (Philadelphia, Pennsylvania, USA)", J. FIELD, M. HICKS (editors), ACM, January 2012, p. 191–202.

[31] S. BASU, T. BULTAN, M. OUEDERNI. *Synchronizability for Verification of Asynchronously Communicating Systems*, in "Proceedings of the 13th International Conference on Verification, Model Checking, and Abstract Interpretation VMCAI'2012 (Philadelphia, PA, USA)", V. KUNCAK, A. RYBALCHENKO (editors), LNCS, Springer, January 2012, vol. 7148, p. 56–71.

[32] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Abstraction-Based Malware Analysis Using Rewriting and Model Checking*, in "Proceedings of the 17th European Symposium on Research in Computer Security ESORICS'2012 (Pisa, Italy)", S. FORESTI, M. YUNG, F. MARTINELLI (editors), LNCS, Springer, September 2012, vol. 7459, p. 806–823.

[33] A. F. E. BELINFANTE. *JTorX: A Tool for On-Line Model-Driven Test Derivation and Execution*, in "Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems TACAS'2010 (Paphos, Cyprus)", Lecture Notes in Computer Science, Springer Verlag, March 2010, vol. 6015, p. 266–270.

[34] F. BOYER, O. GRUBER, G. SALAÜN. *Specifying and Verifying the* SYNERGY *Reconfiguration Protocol with LOTOS NT/CADP*, in "Proceedings of the 17th International Symposium on Formal Methods FM'2011 (Limerick, Ireland)", M. BUTLER, W. SCHULTE (editors), LNCS, Springer, June 2011, vol. 6664, p. 103–117.

[35] F. CAPPELLO, E. CARON, M. DAYDÉ, F. DESPREZ, E. JEANNOT, Y. JEGOU, S. LANTERI, J. LEDUC, N. MELAB, G. MORNET, R. NAMYST, P. PRIMET, O. RICHARD. *Grid'5000: A Large Scale, Reconfigurable, Controlable and Monitorable Grid Platform*, in "Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing GRID'2005 (Seattle, USA)", IEEE/ACM, November 2005, p. 99–106.

[36] D. CHAMPELOVIER, X. CLERC, H. GARAVEL, Y. GUERTE, C. MCKINTY, V. POWAZNY, F. LANG, W. SERWE, G. SMEDING. *Reference Manual of the LOTOS NT to LOTOS Translator (Version 5.7)*, November 2012, Inria/VASY, 153 pages.

[37] E. M. CLARKE, E. A. EMERSON, A. P. SISTLA. *Automatic Verification of Finite-State Concurrent Systems using Temporal Logic Specifications*, in "ACM Transactions on Programming Languages and Systems", April 1986, vol. 8, n^{o} 2, p. 244–263.

[38] R. DE NICOLA, F. W. VAANDRAGER. *Action versus State Based Logics for Transition Systems*, LNCS, Springer Verlag, 1990, vol. 469, p. 407–419.

[39] M. FRAPPIER, R. SAINT-DENIS. *EB^3: An Entity-Based Black-Box Specification Method for Information Systems*, in "Software and System Modeling", 2003, vol. 2, n^{o} 2, p. 134-149.

[40] O. GANEA, F. POP, C. DOBRE, V. CRISTEA. *Specification and Validation of a Real-Time Simple Parallel Kernel for Dependable Distributed Systems*, in "Proceedings of the 3rd International Conference on Emerging Intelligent Data and Web Technologies EIDWT'2012 (Bucharest, Romania)", IEEE Computer Society, September 2012, p. 320–325.

[41] 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)", S. T. VUONG (editor), North Holland, December 1989, p. 147–162.

[42] 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)", Berlin, B. STEFFEN (editor), LNCS, Springer, March 1998, vol. 1384, p. 68–84, Full version available as Inria Research Report RR-3352.

[43] H. GARAVEL, 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)", M. KIM, B. CHIN, S. KANG, D. LEE (editors), Kluwer Academic Publishers, August 2001, p. 377–392, Full version available as Inria Research Report RR-4223.

[44] H. GARAVEL, F. LANG, R. MATEESCU. *Compiler Construction using LOTOS NT*, in "Proceedings of the 11th International Conference on Compiler Construction CC 2002 (Grenoble, France)", N. HORSPOOL (editor), LNCS, Springer, April 2002, vol. 2304, p. 9–13.

[45] H. GARAVEL, R. MATEESCU, D. BERGAMINI, A. CURIC, N. DESCOUBES, C. JOUBERT, I. SMARANDACHE-STURM, G. STRAGIER. *DISTRIBUTOR and BCG_MERGE: Tools for Distributed Explicit State Space Generation*, in "Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems TACAS'2006 (Vienna, Austria)", H. HERMANNS, J. PALBERG (editors), LNCS, Springer, March–April 2006, vol. 3920, p. 445–449.

[46] H. GARAVEL, R. MATEESCU, 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)", Berlin, M. B. DWYER (editor), LNCS, Springer, May 2001, vol. 2057, p. 217–234, Revised version available as Inria Research Report RR-4341 (December 2001).

[47] H. GARAVEL, W. SERWE. *State Space Reduction for Process Algebra Specifications*, in "Theoretical Computer Science", February 2006, vol. 351, n° 2, p. 131–145.

[48] H. GARAVEL, J. SIFAKIS. *Compilation and Verification of LOTOS Specifications*, in "Proceedings of the 10th International Symposium on Protocol Specification, Testing and Verification (Ottawa, Canada)", L. LOGRIPPO, R. L. PROBERT, H. URAL (editors), North Holland, June 1990, p. 379–394.

[49] D. GUCK, T. HAN, J.-P. KATOEN, M. R. NEUHÄUSSER. *Quantitative Timed Analysis of Interactive Markov Chains*, in "Proceedings of the 4th NASA International Symposium on Formal Methods (Norfolk, VA, USA)", A. GOODLOE, S. PERSON (editors), LNCS, Springer, April 2012, vol. 7226, p. 8–23.

[50] M. HENNESSY, R. MILNER. *Algebraic Laws for Nondeterminism and Concurrency*, in "Journal of the ACM", 1985, vol. 32, p. 137–161.

[51] C. JARD, T. JÉRON. *TGV: Theory, Principles and Algorithms — A Tool for the Automatic Synthesis of Conformance Test Cases for Non-Deterministic Reactive Systems*, in "Springer International Journal on Software Tools for Technology Transfer (STTT)", August 2005, vol. 7, n° 4, p. 297–315.

[52] N. KOKASH, C. KRAUSE, E. DE VINK. *Reo + mCRL2: A Framework for Model-Checking Dataflow in Service Compositions*, in "Formal Aspects of Computing", March 2012, vol. 24, n° 2, p. 187–216.

[53] E. LANTREIBECQ, W. SERWE. *Model Checking and Co-simulation of a Dynamic Task Dispatcher Circuit Using CADP*, in "Proceedings of the 16th International Workshop on Formal Methods for Industrial Critical Systems FMICS 2011 (Trento, Italy)", G. SALAÜN, B. SCHÄTZ (editors), LNCS, Springer, August 2011, vol. 6959, p. 180–195.

[54] H.-V. LUONG, A.-L. COURBIS, T. LAMBOLAIS, T. L. PHAN. *IDCM : un outil d'analyse de composants et d'architectures dédié à la construction incrémentale*, in "Actes des 11èmes Journées Francophones sur les Approches Formelles dans l'Assistance au Développement de Logiciels AFADL'2012 (Grenoble : France)", January 2012.

[55] J. MAGEE, J. KRAMER. *Concurrency: State Models and Java Programs*, 2006, Wiley, April 2006.

[56] R. MATEESCU, 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)", J. CUELLAR, T. MAIBAUM, K. SERE (editors), LNCS, Springer, May 2008, vol. 5014, p. 148–164.

[57] J. PARROW, P. SJÖDIN. *Designing a Multiway Synchronization Protocol*, in "Computer Communications", 1996, vol. 19, n^{o} 14, p. 1151–1160.

[58] B. THEELEN, J.-P. KATOEN, W. HAO. *Model Checking of Scenario-Aware Dataflow with CADP*, in "Proceedings of Design, Automation & Test in Europe Conference & Exhibition DATE'2012 (Dresden, Germany)", W. ROSENSTIEL, L. THIELE (editors), IEEE Press, March 2012, p. 653–658.

[59] J. TRETMANS. *Model Based Testing with Labelled Transition Systems*, in "Formal Methods and Testing", LNCS, Springer, 2008, vol. 4949, p. 1–38.

[60] K. J. TURNER, K. L. L. TAN. *Rigorous Development of Composite Grid Services*, in "Journal of Network and Computer Applications", 2012, vol. 35, p. 1304–1316.

[61] F. VAN DER BERG. *DFTCalc - Calculating DFTs using Lotos NT*, University of Twente, 2012.

[62] H. ZHAO, W. WANG, J. SUN, Y. WEI. *Research on Formal Modeling and Verification of BPEL-based Web Service Composition*, in "Proceedings of the 11th International Conference on Computer and Information Science ICIS'2012 (Shanghai, China)", IEEE, June 2012.

[63] N. ZHAR, M. AIT ALI, M. ELEULDJ, A. RAJI. *A Specific-domain Design Tool for FPGA-based Image and Video Processing System*, in "International Journal of Computer Applications", October 2012, vol. 56, n^{o} 11, p. 6–21.