IN PARTNERSHIP WITH:
CNRS

Institut polytechnique de Grenoble

Université Joseph Fourier (Grenoble)

Activity Report 2013

Project-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

Project-Team CONVECS

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

Creation of the Team: 2012 January 01*, updated into Project-Team:* 2014 January 01.

1. Members

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

Faculty Member
Gwen Salaün [Grenoble INP, Associate Professor, HdR]

Engineers
Eric Léo [Inria, granted by Conseil Général de l'Isère]
Soraya Arias [Inria, Engineer, from Jun 2013]

PhD Students
Rim Abid [Univ. Grenoble I]
Hugues Evrard [Inria, granted by Caisse des Dépôts et Consignations]
Fatma Jebali [Inria, granted by Conseil Général de l'Isère]
Abderahman Kriouile [CIFRE with STMicroelectronics]
Raquel Oliveira [Univ. Grenoble I]

Post-Doctoral Fellows
Jingyan Jourdan-Lu [Inria, granted by Conseil Général de l'Isère]
Lina Ye [Inria, granted by Caisse des Dépôts et Consignations]

Visiting Scientist
Zhen Zhang [Inria, from Sep 2013 until Dec 2013]

Administrative Assistants
Helen Pouchot [Inria, until Apr 2013]
Myriam Etienne [Inria, from May 2013]

Other
Kaoutar Hafdi [Inria, internship, from Feb 2013 until Jul 2013]

2. Overall Objectives

2.1. Overview

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

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

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

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

3. Research Program

3.1. New Formal Languages and their Concurrent Implementations

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

- *Model-based specifications*: these are operational (i.e., constructive) descriptions of systems, usually expressed in terms of processes that execute concurrently, synchronize together and communicate. Process calculi are typical examples of model-based specification languages. The approach we promote is based on LOTOS NT (LNT for short), a formal specification language that incorporates most constructs stemming from classical programming languages, which eases its acceptance by students and industry engineers. LNT [32] 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*) [58], 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 a basis for any forward exploration algorithm. On-the-fly verification tools are complex software artifacts, which must be designed as modularly as possible to enhance their robustness, reduce their development effort, and facilitate their evolution. To achieve such a modular framework, we undertake research in several directions:

- *New interfaces for on-the-fly LTS manipulation*. The current application programming interface (API) for on-the-fly graph manipulation, named OPEN/CAESAR [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 make it possible to take advantage of multi-core processors.

- *New generic components for on-the-fly verification*. The quest for new on-the-fly components for LTS analysis must be pursued, with the goal of obtaining a rich 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,
- *Component-based systems:* Web services, peer-to-peer networks,
- *Databases:* transaction protocols, distributed knowledge bases, stock management,
- *Distributed systems:* virtual shared memory, dynamic reconfiguration algorithms, fault tolerance algorithms, cloud computing,
- *Embedded systems:* air traffic control, avionic systems, medical devices,
- *Hardware architectures:* multiprocessor architectures, systems on chip, cache coherency protocols, hardware/software codesign,
- *Human-machine interaction:* graphical interfaces, biomedical data visualization, plasticity,
- *Security protocols:* authentication, electronic transactions, cryptographic key distribution,
- *Telecommunications:* high-speed networks, network management, mobile telephony, feature interaction detection.

5. Software and Platforms

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 [1]. 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 [47], [46] 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 on-the-fly analysis tools have been developed within the OPEN/CAESAR environment, among which:
  - BISIMULATOR, which checks bisimulation equivalences and preorders,
  - CUNCTATOR, which performs steady-state simulation of continuous-time Markov chains,
  - DETERMINATOR, which eliminates stochastic nondeterminism in normal, probabilistic, or stochastic systems,
  - DISTRIBUTOR, which generates the graph of reachable states using several machines,
  - EVALUATOR, which evaluates MCL formulas,
  - EXECUTOR, which performs random execution,
  - EXHIBITOR, which searches for execution sequences matching a given regular expression,
  - GENERATOR, which constructs the graph of reachable states,
  - PROJECTOR, which computes abstractions of communicating systems,
  - REDUCTOR, which constructs and minimizes the graph of reachable states modulo various equivalence relations,
  - SIMULATOR, XSIMULATOR, and OCIS, which enable interactive simulation, and
  - TERMINATOR, which searches for deadlock states.

[1] http://cadp.inria.fr

- BCG (*Binary Coded Graphs*) is both a file format for storing very large graphs on disk (using efficient compression techniques) and a software environment for handling this format. BCG also plays a key role in CADP as many tools rely on this format for their inputs/outputs. The BCG environment consists of various libraries with their programming interfaces, and of several tools, such as:
  - BCG_CMP, which compares two graphs,
  - BCG_DRAW, which builds a two-dimensional view of a graph,
  - BCG_EDIT, which allows the graph layout produced by BCG_DRAW to be modified interactively,
  - BCG_GRAPH, which generates various forms of practically useful graphs,
  - BCG_INFO, which displays various statistical information about a graph,
  - BCG_IO, which performs conversions between BCG and many other graph formats,
  - BCG_LABELS, which hides and/or renames (using regular expressions) the transition labels of a graph,
  - BCG_MIN, which minimizes a graph modulo strong or branching equivalences (and can also deal with probabilistic and stochastic systems),
  - BCG_STEADY, which performs steady-state numerical analysis of (extended) continuous-time Markov chains,
  - BCG_TRANSIENT, which performs transient numerical analysis of (extended) continuous-time Markov chains, and
  - XTL (*eXecutable Temporal Language*), which is a high level, functional language for programming exploration algorithms on BCG graphs. XTL provides primitives to handle states, transitions, labels, *successor* and *predecessor* functions, etc.

    For instance, one can define recursive functions on sets of states, which allow evaluation and diagnostic generation fixed point algorithms for usual temporal logics (such as HML [51], CTL [36], ACTL [37], etc.) to be defined in XTL.
- PBG (*Partitioned BCG Graph*) is a file format implementing the theoretical concept of *Partitioned LTS* [45] 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 [57] 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, Wendelin Serwe.

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 CONVECS.

The TRAIAN compiler can be freely downloaded from the CONVECS Web site [2].

5.3. The PIC2LNT Translator

Participants: Radu Mateescu, Gwen Salaün [correspondent].

We develop a translator named PIC2LNT from an applied π-calculus (see § 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 [3].

5.4. The PMC Partial Model Checker

Participants: Radu Mateescu, Frédéric Lang.

We develop a tool named PMC (*Partial Model Checker*, see § 6.4), which performs the compositional model checking of dataless MCL formulas on networks of communicating automata described in the EXP language.

PMC can be freely downloaded from the CONVECS Web site [4].

6. New Results

6.1. New Formal Languages and their 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 CONVECS for industrial case studies and applications (see § 6.5) and serves also in university courses on concurrency, in particular at ENSIMAG (Grenoble) and at 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).

[2] http://convecs.inria.fr/software/traian

[3] http://convecs.inria.fr/software/pic2lnt

[4] http://convecs.inria.fr/software/pmc

In 2013, in addition to 15 bug fixes, the following enhancements have been brought to these tools:

- The list of predefined functions that can be generated automatically for list and set types has been enriched, so as to include all operations commonly found in programming languages.
- A new “sorted set” type was added to LNT, where the automatically generated insertion function preserves the invariant that all elements in the set are sorted in ascending order and have a single occurrence.
- The use of range and predicate types has been facilitated, by translating explicit type annotations by type conversions if necessary.
- An implicit type conversion is applied by assignments to a variable; this helps the type-checker to solve overloaded definitions.
- The generated LOTOS and C code has been modified to avoid spurious warning messages from the LOTOS and C compilers.
- The demo examples demo_19, demo_30, and demo_35 have been enhanced with an LNT version.

6.1.2. Translation from LOTOS to Petri nets and C

Participants: Hubert Garavel, Wendelin Serwe.

The LOTOS compilers CAESAR and CAESAR.ADT, which were once the flagship of CADP, now play a more discrete role since LNT (rather than LOTOS) has become the recommended specification language of CADP. Thus, CAESAR and CAESAR.ADT are mostly used as back-end translators for LOTOS programs automatically generated from LNT or other formalisms such as Fiacre, and are only modified when this appears to be strictly necessary.

In 2013, in addition to fixing four bugs, the type checking algorithm of CAESAR and CAESAR.ADT was entirely revised to display less and better messages in case of typing errors, avoiding cascading error messages, e.g., when an undefined variable or constant is used, or when an overloaded function is improperly used in a context where a unique type is expected.

Also, the CAESAR compiler found a new usefulness as a means to easily produce large-size, realistic Petri nets that can be used as benchmarks by the Petri net community. To make this possible, a new option was added to CAESAR to export the BPN (*Basic Petri Net*) file generated from a LOTOS specification. The definition of the BPN format was made more precise by adding semantic constraints. The CAESAR.BDD tool of CADP was enhanced with two new options, one that checks whether a BPN file satisfies all semantic constraints, and another one that converts a BPN file into PNML (*Petri Net Markup Language*) format.

This work has been done in coordination with Fabrice Kordon and Lom-Messan Hillah (UPMC/LIP6, Paris, France) for the MCC (*Model Checking Contest*) workshop [5]. H. Garavel was in charge of redesigning the model forms used for this contest. One Petri net generated using CAESAR was selected as a benchmark for MCC’2013 and five Petri nets generated using CAESAR have been submitted to MCC’2014.

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. Despite a substantial body of theoretical work in this area, only a few verification tools have been designed for analysing π-calculus specifications automatically. Our objective is to provide analysis features for the π-calculus by reusing the verification technology available for process algebras without mobility. For this purpose, we extended the original polyadic π-calculus with the data types and functions of LNT. This yields a general-purpose applied π-calculus, which is suitable for specifying mobile value-passing concurrent systems belonging to various application domains. Our approach is based on a novel translation from the finite control fragment of π-calculus to LNT, making possible the analysis of applied π-calculus specifications using all verification tools of CADP. This translation is fully automated by the PIC2LNT translator (see § 5.3).

[5] http://mcc.lip6.fr

In 2013, we continued our work on the applied π-calculus and its translation to LNT. This resulted in a new version PIC2LNT 3.0 of the tool, which fixes several bugs and brings the following improvements:

- A bounded replication operator was added to the language, which expresses the parallel execution of a fixed number of π-calculus agents. This operator is translated into LNT by instantiating the appropriate number of corresponding processes.
- A type Chan representing channel names was implemented, which can be freely combined with ordinary data types. This increases the versatility of the language by allowing, e.g., the definition of agents parametrized by sets of channel names.
- Several options were added to the tool for enhancing its ergonomy and tuning the state space generation (specify the set of private channels that can be created, generate the state space of a particular agent).

A paper describing this work has been published 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 considered a translation from the EB3 language [39] for information systems to LNT. EB3 has a process algebraic flavor, but has the particularity to contain so-called *attribute functions*, whose semantics depend on the history of events. We have proposed a formal translation scheme, which ensures the strong equivalence between the LTSs corresponding to an EB3 specification and to the LNT code generated. A prototype translator has been developed at University Paris-Est Créteil, which enables EB3 specifications to be formally verified using CADP.

In 2013, a paper has been published in an international conference [19].

6.1.5. Coverage Analysis for LNT

Participants: Gwen Salaün, Lina Ye.

In the classic verification setting, the designer has a specification of a system in a value-passing process algebra, a set of temporal properties to be verified on the corresponding LTS model, and a data set of examples (test cases) 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.

In 2013, we proposed a new framework for debugging value-passing process algebraic specifications by means of coverage analysis and we illustrated our approach with LNT. We define several coverage notions before showing how to instrument the specification without affecting its original behavior. Our approach helps the specifier to find dead code, ill-formed conditional structures, and other errors in the specification, but also to improve the quality of a data set of examples used for validation purposes. We have implemented a prototype tool, named CAL, for automating the verification of coverage analysis, and we applied it to several real-world case studies in different application areas. A paper has been submitted to an international conference.

6.1.6. Other Compiler Developments

Participants: Soraya Arias, Hubert Garavel, Frédéric Lang, Wendelin Serwe.

- In co-operation with Jérôme Hugues (ISAE, Toulouse), we investigated the translation of AADL (*Architecture Analysis and Design Language*) into LNT. An AADL example was manually tackled, leading to the conclusion that LNT could be a suitable target language for translating a large fragment of AADL.

In co-operation with Holger Hermanns (Saarland University, Germany) and Joost-Pieter Katoen (RWTH Aachen, Germany), we prepared a contribution for the AADL standardization committee to detail semantics issues of the GSPN (*Generalized Stochastic Petri Nets*) model.

- We continued our work on the FLAC tool, which translates the Fiacre intermediate language into LOTOS to enable verification using CADP. In 2013, we eliminated spurious compilation warnings, we removed the definitions of integer operations *div* and *mod*, which have been added to a standard LOTOS library, and we improved the encoding of integer numbers. These changes have led to revisions 76 to 79 of the FLAC code, which is available on the development forge dedicated to Fiacre compilers [6].
- In co-operation with Holger Hermanns, we started studying the PseuCo language that is being defined and implemented at Saarland University. Developed from an educational perspective as a means to teach concurrency theory to bachelor students, PseuCo combines features from Java and Go, the language promoted by Google for concurrent programming. PseuCo supports both message-passing and shared-memory concurrency in a way that is easy to use and that can readily be transferred to Java, Go, or other mainstream languages. PseuCo has been awarded with the 2013 German national "*Preis des Fakultätentages Informatik*" for its innovative role in undergraduate education.

  In 2013, we undertook the manual translation of various PseuCo sample programs into LNT and started enhancing LNT with features that would enable automated PseuCo-to-LNT translation. We also reviewed a PseuCo-to-CCS translator recently developed at Saarland University and wrote an evaluation report for this software.

6.2. Parallel and Distributed Verification

6.2.1. Manipulation of Partitioned LTSs

Participants: Hubert Garavel, Radu Mateescu, Wendelin Serwe.

For distributed verification, CADP provides the PBG format, which implements the theoretical concept of *Partitioned LTS* [45] and provides a unified access to an LTS distributed over a set of remote machines.

In 2013, we continued the development of the prototype tool 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 also developed another prototype tool, named PBG_INVERT, which changes the storage of the transitions of a partitioned LTS, transforming a partitioned LTS where each fragment stores the transitions leading to the states of the fragment (as generated by DISTRIBUTOR) into a partitioned LTS where each fragment stores the transitions going out from the states of the fragment. Adding this transformation step yields a reduction of up to 25% of the overall execution time, when verifying the partitioned LTS with PBG_OPEN. We experimented all these tools on the Grid'5000 computing infrastructure [31] 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 [13].

6.2.2. Distributed Code Generation for LNT

Participants: Hugues Evrard, Frédéric Lang.

[6] http://gforge.enseeiht.fr/projects/fiacre-compil

Rigorous development and prototyping of a distributed verification algorithm in LNT involves the automatic generation of a distributed implementation. For the latter, a protocol realizing process synchronization is required. 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 2013, we formally modelled and verified several existing synchronization protocols. This revealed an error in one of them, which led to a publication in an international conference [12]. Based on this study, we selected a suitable protocol and adapted it to the LNT synchronization operators.

Using this protocol, we developed a prototype distributed code generator, taking as input the model of a distributed system, described as a set of LNT processes and their parallel composition written in EXP. The LNT.OPEN and CAESAR tools are used to obtain the sequential implementation of each LNT process, and the EXP.OPEN tool is used to compute the possible interactions between processes. Then, our prototype generates the corresponding implementation of the distributed synchronization protocol and all necessary glue code between processes and the protocol. Our prototype automatically performs all these steps, such that a complete and runnable distributed implementation can easily be obtained from the original model.

So far, our prototype manages synchronizations with no data or data of enumerated types only, in which case the implementation checks that data values and types match before allowing a synchronization.

6.3. Timed, Probabilistic, and Stochastic Extensions

Participants: Hubert Garavel, Frédéric Lang, Radu Mateescu.

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 2013, we did the following steps to progress our agenda of bridging the gap between functional verification and quantitative evaluation:

- We equipped CADP with a new tool named BCG_CMP, which enables to compare quantitative models modulo probabilistic and stochastic variants of strong bisimulation and branching bisimulation. Such comparison relations were not available in the BISIMULATOR tool that already existed in CADP.
- We investigated the feasibility of creating interconnections between mainstream verification tools for probabilistic and stochastic systems. In a first step, we focused on the DTMC (*Discrete-Time Markov Chain*) model and on three mainstream tools: CADP (Grenoble), MRMC (Aachen), and PRISM (Birmingham-Oxford).

  We developed translation tools to perform conversions between the various formats of these tools (".aut" and ".bcg" for CADP, ".tra/.sta/.lab" for MRMC, ".pm" and ".tra/.sta/.lab" for PRISM). So doing, we reported one bug in MRMC and five minor issues in PRISM. By discussing with Dave Parker (University of Birmingham), we contributed to the introduction in PRISM 4.1 of two new options "-importmodel" and "-exportmodel" that greatly simplify exchanges of models between PRISM and other tools.

We developed a generator of random DTMCs in CADP, MRMC, and PRISM formats, and undertook the construction of a collection of DTMCs, which we used to compare the performance and scalability of CADP and PRISM.

- We started to investigate the evaluation of temporal logic properties on extended DTMCs, in which transitions are labeled with probabilities and optional actions. For this purpose, we developed a new prototype XTL library (consisting of XTL and C code) encoding the PCTL (*Probabilistic CTL*) temporal logic [50]. This new PCTL library enables the specifier to combine data-based, discrete-time, and probabilistic properties of DTMCs in a uniform way.

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 [26]. 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 2013, we extended the approach to handle fairness operators of alternation depth two, and we conducted new experiments. This resulted in a new version of the PMC prototype tool (see § 5.4) supporting all features of the input language of EXP.OPEN 2.1. An article has been published in an international journal [5].

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 an implementation is conform to a formal model written in LNT. Our approach is inspired by the theory of conformance testing [68], as implemented for instance in TGV [53] and JTorX [30].

We developed two prototype tools. The first tool implements a dedicated OPEN/CAESAR-compliant compiler for the particular asymmetric synchronous product of the model and the test purpose, and uses slightly extended generic components for graph manipulation (τ-compression, τ-confluence reduction, determinization) and resolution of Boolean equation systems. 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. The principal advantage of our approach compared to existing tools is the use of LNT for test purposes, facilitating the manipulation of data values.

In 2013, we continued the development of these tools, with a focus on reducing execution time. We also implemented a prototype tool to extract from a complete test graph one or all test cases of minimal depth. We experimented with these tools on two case-studies, namely the ACE coherence protocol (see § 6.5.1) and the EnergyBus (see § 6.5.5).

6.4.3. Equivalence Checking

Participant: Frédéric Lang.

Equivalence relations can be used for verification in two complementary ways: for the minimization of an LTS and the comparison of two LTSs.

In 2013, we worked along the following lines:

- We added observational equivalence (following a request from LAAS-CNRS) as well as divergence-sensitive branching bisimulation (together with its stochastic and probabilistic variants) in BCG_MIN.
- We improved the speed of BCG_MIN in the case of branching reduction applied to a graph with a high branching factor and many internal transitions, by correcting a function that has a quadratic complexity instead of a linear one.
- We added the new tool BCG_CMP, which takes as input two BCG graphs and checks whether they are equivalent modulo a relation chosen among strong and branching bisimulation (and their stochastic and probabilistic variants), divergence-sensitive branching bisimulation, or observational equivalence. BCG_CMP checks equivalence using the partition-refinement algorithm of BCG_MIN. We compared BCG_CMP and BISIMULATOR on the VLTS benchmark suite [7], showing that BCG_CMP is generally slightly less efficient than BISIMULATOR for comparisons yielding a FALSE result, but much more efficient than BISIMULATOR for comparisons yielding a TRUE result.
- The new tool BCG_CMP as well as the new equivalence relations added to BCG_MIN have been added to the EUCALYPTUS graphical user interface and to the SVL scripting language.

6.4.4. *Other Software Developments*

The OPEN/CAESAR environment was enhanced with a new generic library (named CAESAR_CACHE_1) for manipulating hierarchical caches, with 15 built-in replacement strategies and the possibility to define new ones.

We also maintained the CADP toolbox, taking into account the feedback received from numerous users in the world. In addition to fixing 41 bugs, we evolved CADP to support the latest versions of Windows, Cygwin, Mac OS X, and their corresponding C compilers. The documentation for installing CADP has been updated and shortened. Finally, support for Sparc, Itanium, and PowerPC processors was dropped at the end of 2013 based on the observation that these architectures are almost no longer used among the CADP user community.

6.5. Real-Life Applications and Case Studies

6.5.1. *ACE Cache Coherency Protocol*

Participants: Abderahman Kriouile, Radu Mateescu, Wendelin Serwe.

In the context of a CIFRE convention with STMicroelectronics, we studied 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 techniques grows exponentially. As an alternative, we study formal verification.

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 3200 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 (currently, a representative subset of 15 operations), including an abstract operation that represents any other ACE operation. We handled the global requirements of the ACE specification using a constraint oriented programming style, i.e., by representing each global requirement as a dedicated process observing the global behaviour and inhibiting incorrect executions.

[7] http://cadp.inria.fr/resources/vlts

In a second step, we generated for several configurations the corresponding LTS (up to 100 million states and 350 million transitions). We wrote two liveness properties in MCL expressing that each read (respectively write) transaction is executed until its termination. We also wrote two properties expressing cache coherence and data integrity. This required to transform state-based properties into action-based properties, by adding information about the cache state to actions executed by the cache. For all considered configurations, we checked these properties using parametric SVL scripts (about 100 lines) and EVALUATOR. For some scenarios without the processes representing the global requirements, EVALUATOR generated counterexamples for the cache coherence and data integrity. We are currently using these counterexamples to derive test cases for the architecture under design at STMicroelectronics.

This work led to publications [21], [15].

6.5.2. Choreography-based Communicating Systems

Participants: Radu Mateescu, Gwen Salaün, Lina Ye, Kaoutar Hafdi.

Choreographies are contracts specifying interactions among a set of services from a global point of view. These contracts serve as reference for the further development steps of the distributed system. Therefore, their specification and analysis is crucial to avoid issues (e.g., deadlocks) that may induce delays and additional costs if identified lately in the design and development process.

In 2013, we have obtained the following results:

- In collaboration with Meriem Ouederni (University of Toulouse) and Tevfik Bultan (University of California at Santa Barbara), we have proposed a branching definition of the synchronizability property, which identifies systems whose interaction behavior remains the same when asynchronous communication is replaced with synchronous communication. We have also shown how these results can be used for checking the compatibility of a set of asynchronously communicating components [17].
- In collaboration with Matthias Güdemann (Systerel), we have defined sufficient conditions for checking the repairability property, which indicates whether realizability can be enforced for choreography-based communicating systems using distributed controllers. A paper has been submitted to an international conference.
- We have proposed an approach for computing the degree of parallelism of BPMN processes using model checking techniques. A paper has been submitted to an international conference.
- In collaboration with Pascal Poizat (University of Paris Ouest Nanterre), we have been working on the development of the VerChor platform, which aims at assembling all the verification techniques and tools automating the analysis of choreography specifications [14].

6.5.3. Deployment and Reconfiguration Protocols for Cloud Applications

Participants: Rim Abid, Gwen Salaün.

We collaborated with Noël de Palma and Fabienne Boyer (University Joseph Fourier), Xavier Etchevers and Thierry Coupaye (Orange Labs, Meylan, France) in the field of cloud computing applications, which are complex distributed applications composed of interconnected software components running on distinct virtual machines. Setting up, (re)configuring, and monitoring these applications involves intricate management protocols, which fully automate these tasks while preserving application consistency as well as some key architectural invariants.

In 2013, we focused on the reliability of the self-configuration protocol [22]. This protocol always succeeds in deploying a cloud application, even when facing a finite number of virtual machine or network failures. Designing such highly parallel management protocols is difficult, therefore formal modelling techniques and verification tools were used for validation purposes. These results were accepted for publication in an international conference [11]. Also, an experience export on the verification tasks for such (re)configuration protocols has been published in an international journal [8].

We have also worked on the design and verification of a reconfiguration protocol, where virtual machines interact altogether using a publish-subscribe messaging system. The verification of this protocol with CADP helped to refine several parts of the protocol and correct subtle bugs. These results have been published in an international conference [10]. In collaboration with Francisco Durán (University of Málaga), we have also worked on the design of a variant of this reconfiguration protocol, where the virtual machines interact via FIFO buffers. A paper has been submitted to an international conference.

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 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, in particular model checking and equivalence checking.

In 2013, we studied the existing approaches and languages that address formal modeling and verification of GALS systems. We concluded that the current landscape lacks general-purpose, flexible, and formal representation of GALS systems suitable for efficient verification. To fulfill this requirement, we have designed GRL (*GALS Representation Language*), a language with user-friendly syntax and formal semantics, to efficiently model GALS systems for the purpose of formal verification. GRL targets GALS systems consisting of networks of synchronous systems interacting with their environments and communicating via asynchronous media. GRL draws mainly from two foundations. Regarding asynchronous concurrency, GRL builds upon process calculi (in particular LNT). Thereby, it leverages process calculi expressiveness, versatility, and verification efficiency. Regarding synchronous features, GRL holds a dataflow-oriented model based on the dataflow diagram model (also called block-diagram model). The GRL synchronous model inherits from the simplicity and modularity of the block-diagram model.

We defined the lexical and the abstract syntax of GRL (about 80 grammar rules), its static semantics (about 150 binding, typing, and initialization rules), and its dynamic semantics (about 20 structured operational semantics rules). Using the SYNTAX and LOTOS NT compiler construction technology, we started the development of a prototype translator GRL2LNT (about 8000 lines). The tool currently performs the lexical and syntactic analysis of GRL programs, together with some static semantic checks. A database containing about 30 examples of GRL programs has been constructed and used for non-regression testing of GRL2LNT. A reference manual for GRL (130 pages up to now) containing the definition of the language and its translation to LNT has been written. A paper presenting the GRL language has been submitted to an international conference.

Regarding the analysis of PLC networks by equivalence checking, we defined variants of classic equivalence relations (strong, τ*.a, and branching) for comparing the Mealy machine corresponding to a PLC network with the Moore machine corresponding to its external behaviour. We reformulated the verification problem as the resolution of a Boolean equation system, and we developed a prototype tool, based on the CAESAR_SOLVE_1 library, for the on-the-fly comparison of a Mealy and a Moore machine modulo the strong or the τ*.a equivalences.

6.5.5. *EnergyBus Standard for Connecting Electric Components*

Participants: Hubert Garavel, Wendelin Serwe.

The EnergyBus [8] is an upcoming industrial standard for electric power transmission and management, based on the CANopen field bus. It is developed by a consortium assembling all major industrial players (such as Bosch, Panasonic, and Emtas) in the area of light electric vehicles (LEV); their intention is to ensure interoperability between all electric LEV components. At the core of this initiative is a universal plug integrating a CAN-Bus[9] with switchable power lines. The central and innovative role of the EnergyBus is to manage the safe electricity access and distribution inside an EnergyBus network.

In the framework of the European FP7 project SENSATION (see § 8.2.1.1) a formal specification in LNT of the main EnergyBus protocols is being developed by Alexander Graf-Brill and Holger Hermanns at Saarland University [49], with the active collaboration of CONVECS.

In 2013, CONVECS provided help in modelling using the LNT language and the TGV tool, and enhanced the CADP toolbox to address a number of issues reported by Saarland University. At present, this LNT specification (1670 lines) is used for generating test suites using the TGV tool [53]. The formal modelling prompted for modifications in the EnergyBus standard and the generated test suites revealed three unknown bugs in an industrial CANopen implementation.

6.5.6. *Graphical User-Interfaces and Plasticity*

Participants: Hubert Garavel, Frédéric Lang, Raquel Oliveira.

In the context of the Connexion project (see § 8.1.1.2) and in close co-operation with Gaëlle Calvary, Eric Ceret, and Sophie Dupuy-Chessa (IIHM team of the LIG laboratory), we study the formal description and validation of graphical user-interfaces using the most recent features of the CADP toolbox. The case study assigned to LIG in this project is a prototype graphical user-interface [35] designed to provide human operators with an overview of a running nuclear plant. Contrary to conventional control rooms, which employ large desks and dedicated hardware panels for supervision, this new-generation interface uses standard computer hardware (i.e., smaller screen(s), keyboard, and mouse), thus raising challenging questions on how to best provide synthetic views of status information and alarms resulting from faults, disturbances, or unexpected events in the plant. Another challenge is to introduce plasticity in such interface, so as to enable several supervision operators, including mobile ones outside of the control room, to get accurate information in real time.

In 2013, CONVECS contributed to the following results. Based upon the available information published by EDF, a formal specification in LNT of this new-generation interface was developed (2600 lines). This specification not only encompasses the usual components traditionally found in graphical user-interfaces, but also a model of the physical world (namely, a nuclear reactor with various fault scenarios) and a cognitive model of a human operator in charge of supervising the plant. Also, a few desirable properties of the interface have been expressed in the MCL language of CADP and verified on the LNT model.

So doing, three main difficulties have been faced. The description of the prototype available in the published literature is not exhaustive, which required us to provide those missing details needed to obtain a realistic model. Quite often, we faced a combinatorial explosion in the number of states of the model, which forced us to restrict the complexity of operator behaviour and fault models. Finally, this case study revealed several LNT-specific issues, which triggered enhancements in the LNT language and tools.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Grants with Industry

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

[8] http://www.energybus.org

[9] http://www.can-cia.org

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 (see § 6.5.1), under the supervision of Guilhem Barthes (STMicroelectronics), Christophe Chevallaz (STMicroelectronics), Grégory Faux (STMicroelectronics), Radu Mateescu (CONVECS), Wendelin Serwe (CONVECS), and Massimo Zendri (STMicorelectronics).

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 [10] 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) modelling 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 (see § 6.5.3) are the formal specification of the models, architectures, and protocols (self-deployment, dynamic reconfiguration, self-repair, 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 [11] (*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, 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-LIST, 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. In this project, CONVECS will assist another LIG team, IIHM, in specifying human-machines interfaces formally using the LNT language and in verifying them using CADP (see § 6.5.6).

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].

[10] http://www.opencloudware.org

[11] http://www.cluster-connexion.fr

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 (see § 6.5.4) are the definition of GRL, 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 2013 with the following Inria project-teams:

- OASIS (Inria Sophia-Antipolis – Méditerranée): Eric Madelaine and Ludovic Henrio,
- TRISKELL (Inria Rennes – Bretagne Atlantique): Kevin Corre and Axel Legay,
- MEXICO (Inria Saclay – Île-de-France): Alban Linard.

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

- Gaëlle Calvary and Sophie Dupuy-Chessa (LIG, Grenoble),
- Fabrice Kordon and Lom Messan Hillah (LIP6, Paris),
- Alexandre Hamez and Jérôme Hugues (ISAE, Toulouse),
- Noël De Palma and Fabienne Boyer (LIG, Grenoble),
- Xavier Etchevers (Orange Labs, Meylan),
- Matthias Güdemann (Systerel, Aix-en-Provence),
- Meriem Ouederni (IRIT, Toulouse),
- Pascal Poizat (LIP6, Paris).

H. Garavel, F. Lang, and R. Oliveira attended two training days on the Scade and Scade Display software (given by Luc Coyette, Esterel Technologies) on March 6 and 24, 2013.

8.2. European Initiatives

8.2.1. FP7 Projects

8.2.1.1. SENSATION

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

SENSATION [12] (*Self ENergy-Supporting Autonomous computaTION*) is a European project no. 318490 funded by the FP7-ICT-11-8 programme. It gathers 9 participants: Inria (TRISKELL and CONVECS project-teams), Aalborg University (Denmark), RWTH Aachen and Saarland University (Germany), University of Twente (The Netherlands), 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.

[12] http://sensation-project.eu/

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.

The case study on rescaling video for handheld devices, proposed initially by STMicroelectronics, was abandoned in 2013 after the departure of this partner from the project. Therefore, we oriented our efforts on the EnergyBus case study (see § 6.5.5), in collaboration with Saarland University.

8.2.2. Collaborations with Major European Organizations

The CONVECS project-team is member of the FMICS (*Formal Methods for Industrial Critical Systems*) working group of ERCIM [13]. 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.

H. 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, we had scientific relations in 2013 with several European universities and research centers, including:

- Saarland University (Alexander Graf-Brill and Holger Hermanns),
- RWTH Aachen (Joost-Pieter Katoen),
- Oxford University (Ernst-Moritz Hahn and Marta Kwiatkowska),
- University of Birmingham (Dave Parker),
- Technical University of Eindhoven (Anton Wijs),
- University of Twente (Marieke Huisman and Jaco van de Pol),
- University of Málaga (Francisco Duran and Ernesto Pimentel).

Our partnership with Saarland University was sustained by the Humboldt Forschungspreis received by H. Garavel, who continued his regular visits to Saarland University.

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).

We also had scientific exchanges with Gianfranco Ciardo (University of California at Riverside, USA).

[13] http://fmics.inria.fr

8.4. International Research Visitors

8.4.1. Visits of International Scientists

- Loïg Jezequel (Technical University of München, Germany) visited us on March 4–6, 2013. He gave a talk entitled “*Distributed Cost-Optimal Planning*” on March 4, 2013.
- Zhen Zhang (University of Utah, USA) visited us from September 1st to December 31, 2013.
- The annual CONVECS seminar was held in Col de Porte (France) on November 18–20, 2013. The following invited scientists attended the seminar:
  - Jérôme Hugues (Institute for Space and Aeronautics Engineering, Toulouse) gave on November 18, 2013 a talk entitled “*Model-Based, Model Checking: the Missing Bits*”.
  - Loïg Jezequel (Technical University of München, Germany) gave on November 19, 2013 a talk entitled “*Computation of Summaries using Net Unfoldings*”.
  - Xavier Etchevers (Orange Labs, Meylan, France) gave on November 19, 2013 a talk entitled “*VAMP: Self-Deployment of Arbitrary Applications in the Cloud*”.
  - Fabrice Kordon (LIP6, Paris) gave on November 20, 2013 a talk entitled “*Verification Approaches for Distributed Systems in LIP6/MoVe*”.
  - Zhen Zhang (University of Utah, USA) gave on November 20, 2013 a talk entitled “*Modeling a Fault-Tolerant Wormhole Routing Algorithm using LNT*”.

9. Dissemination

9.1. Scientific Animation

9.1.1. Software Dissemination and Internet Visibility

The CONVECS project-team distributes several software tools: the CADP toolbox (see § 5.1), the TRAIAN compiler (see § 5.2), the PIC2LNT translator (see § 5.3), and the PMC model checker (see § 5.4). In 2013, the main facts are the following:

- We prepared and distributed 11 successive versions (from 2013-a to 2013-l “Zurich”) of CADP.
- We were requested to grant CADP licenses for 934 different computers in the world.
- We released version 3.0 of the PIC2LNT translator from an applied π-calculus to LNT in May 2013.
- We released version 1.0 of the PMC partial model checker for networks of automata in May 2013.

The CONVECS Web site [14] was updated with scientific contents, announcements, publications, etc.

By the end of December 2013, the CADP forum [15], opened in 2007 for discussions regarding the CADP toolbox, had over 270 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 verification of BPMN models with the Alvis modeling language [64], [65], [66]
- The DFTCalc tool for efficient fault tree analysis [29], [28]
- Efficient modeling, generation, and analysis of Markov automata [67]
- Modeling and verification techniques for the incremental development of UML architectures [63]
- Model based design of complex embedded systems [27]
- Model extraction approach to verifying concurrent C programs [40]

[14] http://convecs.inria.fr

[15] http://cadp.inria.fr/forum.html

- Model checking based approach to automatic test suite generation for Web services and BPEL [73]
- The VIP Design graphical language for the design of image and video processing embedded systems [74], [75]
- Active learning of extended finite state machines [69]
- Incremental construction and interoperability analysis of critical systems [56], [55], [62]
- Modeling robot behavior with CCL [54]
- Behavioural verification of distributed components [52]
- Efficient property preservation checking of model refinements [71]
- Efficient operational semantics for EB3 for verification of temporal properties [70]
- Multilevel contracts for trusted components [59]

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

- Formally reasoning on a reconfigurable component-based system [48]
- Assisting refinement in system-on-chip design [60]
- Formal development of control software in the medical systems domain [61]
- Model-driven approach supporting formal verification for Web service composition protocols [38]
- Scalably verifiable cache coherence [72]
- Improved test case generation from UML statecharts [34], [33]

9.1.2. Program Committees

In 2013, 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 is an editorial board member of the Scientific World Journal in the Computer Science subject area.
- G. Salaün is an editorial board member of SOCA (*Springer International Journal on Service Oriented Computing and Applications*).
- G. Salaün is a steering committee member of FACS (*International Symposia on Formal Aspects of Component Software*).
- G. Salaün is a steering committee member of FOCLASA (*International workshops on Foundations of Coordination Languages and Self-Adaptive Systems*).
- G. Salaün was a program committee member for MODELSWARD'2013 / MODA'2013 (*1st International Conference on Model-Driven Engineering and Software Development, Special Session on Model-Driven Software Adaptation*), Barcelona, Spain, February 19–21, 2013.
- G. Salaün was a program committee member for SAC'2013 (*28th Annual ACM Symposium on Applied Computing, Track on Service-Oriented Architectures and Programming*), Coimbra, Portugal, March 18–22, 2013.
- G. Salaün and W. Serwe were program committee members for FSEN'2013 (*5th International Conference on Fundamentals of Software Engineering*), Tehran, Iran, April 24–26, 2013.
- G. Salaün was a program committee member for CBSE'2013 (*16th International ACM Sigsoft Symposium on Component-Based Software Engineering*), Vancouver, Canada, June 18–20, 2013.
- G. Salaün was a program committee member for QASBA'2013 (*2nd International Workshop on Quality Assurance for Service-based Applications*), Lugano, Switzerland, July 15, 2013.
- F. Lang was a program committee member for ETR'2013 (*Ecole d'été Temps-réel*), Toulouse, France, August 26–30, 2013.
- F. Lang was a program committee member for ESOCC'2013 (*European Conference on Service-Oriented and Cloud Computing*), Málaga, Spain, September 11–13, 2013.
- F. Lang and R. Mateescu were program committee members for FMICS'2013 (*18th International Workshop on Formal Methods for Industrial Critical Systems*), Madrid, Spain, September 23–24, 2013.
- G. Salaün was a program committee member for FACS'2013 (*10th International Symposium on Formal Aspects of Component Software*), Nanchang, China, October 28–30, 2013.

9.1.3. Awards and Distinctions

H. Garavel is an invited professor at Saarland University (Germany) as a holder of the Gay-Lussac Humboldt Prize.

9.1.4. Lectures and Invited Conferences

- H. Garavel attended the Dagstuhl Seminar #13051 on "*Software Certification: Methods and Tools*" (Schloss Dagstuhl, Germany, January 27 – February 1st, 2013). He gave a lecture entitled "*A Naive Look at Software Certification Practices – and Proposals for Enhancement*" on January 30, 2013.
- G. Salaün visited the University of Málaga (Spain) from February 2 to March 8, 2013. He gave a talk entitled "*Formal Methods for Cloud Computing Environments*" on February 6, 2013 and a talk entitled "*Verification of Contract-based Communicating Systems*" on February 13, 2013.
- H. Garavel visited RWTH Aachen (Germany) on February 25–28, 2013. He gave a talk entitled "*CAESAR Nets, NTIF, and FIACRE: Better than Slim, and also Faster?*".
- G. Salaün gave a keynote lecture entitled "*Verification of Contract-based Communicating Systems*" at GRAPHITE'2013 (Rome, Italy) on March 24, 2013.
- H. Garavel was invited to the seminar "*25 Years of Combining Compositionality and Concurrency*" (Königswinter, Germany, August 7–9, 2013). He gave a lecture entitled "*25 Years of Compositionality Issues in CADP: An Overview*".
- F. Lang gave a lecture entitled "*CADP: A Toolbox for the Construction and Analysis of Distributed Processes*", followed by a lab session on CADP at ETR'2013 (Toulouse, France) on August 28, 2013.
- F. Lang gave a keynote lecture entitled "*Langage de nouvelle génération pour la modélisation et vérification formelle de systèmes asynchrones*" at JDEV'2013 (Palaiseau, France) on September 4, 2013.
- H. Garavel gave a lecture entitled "*25 Years of Combining Compositionality and Concurrency*" at Saarland University on December 20, 2013.

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 2013, we carried out the following teaching activities:

- G. Salaün is co-responsible for the ISI (*Ingéniérie des Systèmes d'Information*) department of ENSIMAG since September 1, 2011.
- 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 (36 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 second year computer science engineering students of ENSIMAG (18 hours).
- A. Kriouile served as a teaching assistant in a course on "*Conception de circuits et architectures des ordinateurs*", given by Frédéric Pétrot to the first year computer science engineering students of ENSIMAG (27 hours).

- A. Kriouile 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 (36 hours).
- A. Kriouile served as a teaching assistant for a student project "*Projet logiciel en C*", proposed by François Broquedis and Matthieu Chabanas to the first year computer science engineering students of ENSIMAG (20 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 a course on "*Algorithmique parallèle et orientée-objet*" to the second year computer science engineering students of ENSIMAG (36 hours).
- L. Ye gave a course on "*Théorie des langages*" to the first year computer science engineering students of ENSIMAG (10 hours).

9.2.2. Juries

- R. Mateescu was a panel member for Syed Hussein Syed Alwi's PhD thesis, entitled "*Vérification compositionnelle pour la conception sûre de systèmes embarqués*", defended at Université Pierre et Marie Curie, Paris, France, on July 11, 2013.
- G. Salaün was a panel member for Huu Nghia Nguyen's PhD thesis, entitled "*A Symbolic Approach for the Verification and the Testing of Service Choreographies*", defended at Université Paris Sud, Orsay, France, on October 30, 2013.

9.3. Popularization

H. Garavel participated to the committee in charge of organizing the Aerospace Valley series of industrial conferences on formal methods. The second conference [16] [17], devoted to static analysis, held on June 28, 2013 in Toulouse and retransmitted by video-conference in Grenoble, attracted 95 participants from industry and academia.

R. Mateescu was in charge of the scientific organization of the In'Tech seminar entitled "*Formal Validation of Industrial Critical Systems*" held on April 18, 2013 at the Inria Grenoble – Rhône-Alpes research center in Montbonnot. The seminar attracted about 100 participants from academia and industry. F. Lang gave a public demonstration of the CADP tools and R. Mateescu gave a talk entitled "*Formal Modeling and Verification of Concurrent Systems using CADP*".

9.4. Miscellaneous Activities

H. Evrard is a member of the council of the MSTII doctoral school.

H. Evrard and G. Salaün are members of the council of the LIG laboratory.

H. Garavel is a member of the LIG commission in charge of preparing candidates selected for recruitment interviews at CNRS.

H. Garavel is a member of the operational committee of the EMSOC cluster ("Embedded System on Chip") within the "*pôle de compétitivité*" Minalogic.

H. Garavel was a reviewer for various ongoing ANR (*Agence Nationale de la Recherche*) projects evaluated in 2013.

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.

E. Léo and W. Serwe are members of the "*comité de centre*" of Inria Grenoble - Rhône-Alpes.

[16] http://www.inria.fr/centre/grenoble/agenda/forum-methodes-formelles

[17] http://www.inria.fr/centre/grenoble/actualites/fiabilite-des-logiciels-pas-uniquement-pour-les-avions

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

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

H. Garavel is "*chargé de mission*" of the LIG laboratory and responsible for the liaison with Minalogic.

W. Serwe is "*chargé de mission*" of the LIG laboratory for the scientific axis "*Formal Methods, Models, and Languages*".

10. Bibliography

Major publications by the team in recent years

[1] 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

[2] 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

[3] 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, pp. 290-304, http://hal.inria.fr/hal-00726448

Publications of the year

Articles in International Peer-Reviewed Journals

[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", 2013, vol. 15, n^{o} 2, pp. 89-107 [*DOI :* 10.1007/s10009-012-0244-z], http://hal.inria.fr/hal-00715056

[5] F. Lang, R. Mateescu. *Partial Model Checking using Networks of Labelled Transition Systems and Boolean Equation Systems*, in "Logical Methods in Computer Science", October 2013, vol. 9, n^{o} 4, http://hal.inria.fr/hal-00872181

[6] E. Lantreibecq, W. Serwe. *Formal Analysis of a Hardware Dynamic Task Dispatcher with CADP*, in "Science of Computer Programming", 2014, vol. 80, pp. 130-149 [*DOI :* 10.1016/j.scico.2013.01.003], http://hal.inria.fr/hal-00782069

[7] N. D. Mendes, F. Lang, Y.-S. Le Cornec, R. Mateescu, G. Batt, C. Chaouiya. *Composition and abstraction of logical regulatory modules: application to multicellular systems*, in "Bioinformatics", January 2013, vol. 29, n^{o} 6, pp. 749-757 [*DOI :* 10.1093/bioinformatics/btt033], http://hal.inria.fr/hal-00785564

[8] G. Salaün, F. Boyer, T. Coupaye, N. De Palma, X. Etchevers, O. Gruber. *An Experience Report on the Verification of Autonomic Protocols in the Cloud*, in "Innovations in Systems and Software Engineering", April 2013 [*DOI :* 10.1007/s11334-013-0204-0], http://hal.inria.fr/hal-00808565

[9] S. STRUCK, M. GÜDEMANN, F. ORTMEIER. *Efficient Optimization of Large Probabilistic Models*, in "Journal of Systems and Software", April 2013 [*DOI :* 10.1016/J.JSS.2013.03.078], http://hal.inria.fr/hal-00816636

International Conferences with Proceedings

[10] R. ABID, G. SALAÜN, F. BONGIOVANNI, N. DE PALMA. *Verification of a Dynamic Management Protocol for Cloud Applications*, in "11th International Symposium, ATVA 2013", Hanoi, Viet Nam, Dang Van Hung and Mizuhito Ogawa, 2013, vol. 8172, pp. 178-192 [*DOI :* 10.1007/978-3-319-02444-8_14], http://hal.inria.fr/hal-00863262

[11] X. ETCHEVERS, G. SALAÜN, F. BOYER, T. COUPAYE, N. DE PALMA. *Reliable Self-Deployment of Cloud Applications*, in "SAC 2014 - 29th ACM Symposium on Applied Computing", Gyeongju, Korea, Republic Of, March 2014, http://hal.inria.fr/hal-00934042

[12] H. EVRARD, F. LANG. *Formal Verification of Distributed Branching Multiway Synchronization Protocols*, in "FORTE / FMOODS - 2013 IFIP Joint International Conference on Formal Techniques for Distributed Systems (33rd FORTE / 15th FMOODS)", Florence, Italy, D. BEYER, M. BOREALE (editors), Springer, April 2013, http://hal.inria.fr/hal-00818788

[13] H. GARAVEL, R. MATEESCU, W. SERWE. *Génération et manipulation d'espaces d'états distribués avec CADP : expériences sur Grid'5000*, in "Conférence en Parallélisme, Architecture et Système ComPAS'2013", Grenoble, France, 2013, http://hal.inria.fr/hal-00777110

[14] M. GÜDEMANN, P. POIZAT, G. SALAÜN, A. DUMONT. *VerChor: A Framework for Verifying Choreographies*, in "Fundamental Approaches to Software Engineering 2013", Rome, Italy, March 2013 [*DOI :* 10.1007/978-3-642-37057-1_16], http://hal.inria.fr/hal-00806788

[15] A. KRIOUILE, W. SERWE. *Formal Analysis of the ACE Specification for Cache Coherent Systems-on-Chip*, in "FMICS - 18th International Workshop on Formal Methods for Industrial Critical Systems", Madrid, Spain, M. DIERKES, C. PECHEUR (editors), Lecture Notes in Computer Science (LNCS), Springer, 2013, vol. 8187, pp. 108-122, http://hal.inria.fr/hal-00858521

[16] R. MATEESCU, G. SALAÜN. *PIC2LNT: Model Transformation for Model Checking an Applied Pi-Calculus*, in "TACAS - 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems - 2013", Rome, Italy, N. PITERMAN, S. SMOLKA (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7795, pp. 192-198 [*DOI :* 10.1007/978-3-642-36742-7_14], http://hal.inria.fr/hal-00805533

[17] M. OUEDERNI, G. SALAÜN, T. BULTAN. *Compatibility Checking for Asynchronously Communicating Software*, in "FACS 2013", Nanchang, China, October 2013, http://hal.inria.fr/hal-00913665

[18] M. OUEDERNI, G. SALAÜN, J. CÁMARA, E. PIMENTEL. *Comparator: A Tool for Quantifying Behavioural Compatibility*, in "FASE 2014 - 17th International Conference on Fundamental Approaches to Software Engineering", Grenoble, France, April 2014, http://hal.inria.fr/hal-00934057

[19] D. VEKRIS, F. LANG, C. DIMA, R. MATEESCU. *Translating EB3 to LNT for verification with CADP*, in "iFM - 10th International Conference on integrated Formal Methods - 2013", Turku, Finland, June 2013, http://hal.inria.fr/hal-00768310

[20] L. YE, P. DAGUE, F. NOUIOUA. *Predictability Analysis of Distributed Discrete Event Systems*, in "52nd IEEE Conference on Decision and Control", Florence, Italy, December 2013, pp. 5009-5015, http://hal.inria.fr/hal-00919434

National Conferences with Proceedings

[21] A. KRIOUILE, W. SERWE. *Analyse formelle du protocole ACE : cohérence de caches des systèmes sur puce*, in "École d'été Temps-Réel 2013", Toulouse, France, August 2013, pp. 130-133, http://hal.inria.fr/hal-00876665

Scientific Books (or Scientific Book chapters)

[22] G. SALAÜN, X. ETCHEVERS, N. DE PALMA, F. BOYER, T. COUPAYE. *Verification of a Self-configuration Protocol for Distributed Applications in the Cloud*, in "Assurances for Self-Adaptive Systems", J. CAMARA, R. DE LEMOS, C. GHEZZI, A. LOPES (editors), Springer, January 2013 [*DOI :* 10.1007/978-3-642-36249-1_3], http://hal.inria.fr/hal-00806914

Books or Proceedings Editing

[23] M. R. MOUSAVI, G. SALAÜN (editors). , *Preface: Special Section on Foundations of Coordination Languages and Software Architectures (Selected Papers from FOCLASA'10)*, Elsevier, February 2014, 2 p. , http://hal.inria.fr/hal-00919799

[24] G. SALAÜN, B. SCHÄTZ (editors). , *Preface: Special Section on Formal Methods for Industrial Critical Systems (Selected Papers from FMICS'11)*, Elsevier, February 2014, 2 p. , http://hal.inria.fr/hal-00919803

References in notes

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

[26] 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, pp. 398–407

[27] L. APVRILLE. , *Model-Based Design of Complex Embedded Systems*, University of Nice Sophia-Antipolis, November 2012, Habilitation à Diriger les Recherches

[28] F. ARNOLD, A. BELINFANTE, F. V. DER BERG, D. GUCK, M. STOELINGA. , *DFTCalc: a tool for efficient fault tree analysis (extended version)*, Centre for Telematics and Information Technology, University of Twente, Enschede, July 2013, n^{o} TR-CTIT-13-13

[29] F. ARNOLD, A. BELINFANTE, F. V. DER BERG, D. GUCK, M. STOELINGA. *DFTCalc: A Tool for Efficient Fault Tree Analysis*, in "Proceedings of the 32nd International Conference on Computer Safety, Reliability, and Security SAFECOMP'2013 (Toulouse, France)", F. BITSCH, J. GUIOCHET, M. KAÂNICHE (editors), Lecture Notes in Computer Science, Springer Verlag, September 2013, vol. 8153, pp. 293–301

[30] A. 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, pp. 266–270

[31] 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, pp. 99–106

[32] 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, 153 p. , Inria/VASY

[33] V. CHIMISLIU, F. WOTAWA. *Improving Test Case Generation from UML Statecharts by Using Control, Data and Communication Dependencies*, in "Proceedings of the 13th International Conference on Quality Software QSIC'2013 (Nanjing, China)", IEEE Computer Society Press, July 2013, pp. 125–134

[34] V. CHIMISLIU, F. WOTAWA. *Using Dependency Relations to Improve Test Case Generation from UML Statecharts*, in "Proceedings of the IEEE 37th Annual Computer Software and Applications Conference Workshops COMPSACW'2013 (Kyoto, Japan)", IEEE Computer Society Press, July 2013, pp. 71–76

[35] F. CHÉRIAUX, D. GALARA, M. VIEL. *Interfaces for Nuclear Power Plant Overview*, in "Proceedings of the 8th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies NPIC & HMIT 2012 (San Diego, California, USA)", American Nuclear Society, July 2012

[36] 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, pp. 244–263

[37] R. DE NICOLA, F. W. VAANDRAGER. , *Action versus State Based Logics for Transition Systems*, Lecture Notes in Computer Science, Springer Verlag, 1990, vol. 469, pp. 407–419

[38] C. DUMEZ, M. BAKHOUYA, J. GABER, M. WACK, P. LORENZ. *Model-Driven Approach Supporting Formal Verification for Web Service Composition Protocols*, in "Journal of Network and Computer Applications", July 2013, vol. 36, n^{o} 4, pp. 1102–1115

[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, pp. 134-149

[40] M. GALLARDO, C. JOUBERT, P. MERINO, D. SANÁN. *A model-extraction approach to verifying concurrent C programs with CADP*, in "Science of Computer Programming", March 2012, vol. 77, n^{o} 3, pp. 375—392

[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, pp. 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), Lecture Notes in Computer Science, Springer Verlag, March 1998, vol. 1384, pp. 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, pp. 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), Lecture Notes in Computer Science, Springer Verlag, April 2002, vol. 2304, pp. 9–13

[45] 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), Lecture Notes in Computer Science, Springer Verlag, May 2001, vol. 2057, pp. 217–234, Revised version available as Inria Research Report RR-4341 (December 2001)

[46] H. GARAVEL, W. SERWE. *State Space Reduction for Process Algebra Specifications*, in "Theoretical Computer Science", February 2006, vol. 351, n^{o} 2, pp. 131–145

[47] 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, pp. 379–394

[48] N. GASPAR, L. HENRIO, E. MADELAINE. *Formally Reasoning on a Reconfigurable Component-Based System - A Case Study for the Industrial World*, in "Proceedings of the 10th International Symposium on Formal Aspects of Component Software FACS'13 (Nanchang, China)", Lecture Notes in Computer Science, Springer Verlag, October 2013

[49] A. GRAF-BRILL. , *Model-based Testing Approaches for the EnergyBus*, Department of Computer Science, Faculty of Natural Sciences and Technology I, Saarland University, October 2013

[50] H. HANSSON, B. JONSSON. *A Logic for Reasoning about Time and Reliability*, in "Formal Aspects of Computing", 1994, vol. 6, pp. 102–111

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

[52] L. HENRIO, E. MADELAINE. *Behavioural Verification of Distributed Components*, in "Proceedings of the 6th Interaction and Concurrency Experience ICE'2013 (Florence, Italy)", June 2013

[53] 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^{o} 4, pp. 297–315

[54] K. KULAKOWSKI, T. SZMUC. *Modeling Robot Behavior with CCL*, in "Proceedings of the 3rd International Conference on Simulation, Modeling, and Programming for Autonomous Robots SIMPAR'2012 (Tsukuba, Japan)", Lecture Notes in Artificial Intelligence, Springer Verlag, November 2012, vol. 7628, pp. 40–51

[55] T. LAMBOLAIS, A.-L. COURBIS, H.-V. LUONG, T.-L. PHAN. *Interoperability Analysis of Systems*, in "Proceedings of the 18th IFAC World Congress (Milano, Italy)", IFAC, August 2011, vol. 18, pp. 7879–7884

[56] H.-V. LUONG. , *Construction incrémentale de spécifications de systèmes critiques intégrant des procédures de vérification*, University Paul Sabatier, Toulouse, October 2010

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

[58] 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), Lecture Notes in Computer Science, Springer Verlag, May 2008, vol. 5014, pp. 148–164

[59] M. MESSABIHI, P. ANDRÉ, C. ATTIOGBÉ. *Multilevel Contracts for Trusted Components*, in "Proceedings of the International Workshop on Component and Service Interoperability WCSI'10 (Málaga, Spain)", J. CÁMARA, C. CANAL, G. SALAÜN (editors), EPTCS, June 2010, vol. 37, pp. 71–85

[60] H. MOKRANI, R. AMEUR-BOULIFA, E. ENCRENAZ-TIPHÈNE. *Assisting Refinement in System-on-Chip Design*, in "Proceedings of the 2013 Forum on Specification and Design Languages FDL'2013 (Paris, France)", IEEE Computer Society Press, September 2013, pp. 1–6

[61] A. OSAIWERAN. , *Formal Development of Control Software in the Medical Systems Domain*, Eindhoven University of Technology, December 2012

[62] T.-L. PHAN, A.-L. COURBIS, T. LAMBOLAIS, T. LIBOUREL. , *Incremental Construction of Architectural Specification Supported by Behavioral Verification*, LGI2P, Ecole des Mines d'Alès, June 2012, n^{o} RR12/Lab/002

[63] T.-L. PHAN. , *Modeling and verification techniques for incremental development of UML architectures*, July 2013

[64] M. SZPYRKA, N. GRZEGORZ, L. ANTONI, K. KRZYSZTOF. *Proposal of Formal Verification of Selected BPMN Models with Alvis Modeling Language*, in "Proceedings of the 5th International Symposium on Intelligent Distributed Computing IDC'2011 (Delft, The Netherlands)", F. BRAZIER, K. NIEUWENHUIS, G. PAVLIN, M. WARNIER, C. BADICA (editors), Springer Verlag, October 2011, pp. 243–248

[65] M. SZPYRKA, P. MATYASIK, R. MRÓWKA, L. KOTULSKI. *Communication with Environment in Alvis Models*, in "International Journal of Electronics and Telecommunications", 2012, vol. 58, n^{o} 3, pp. 247–254

[66] M. SZPYRKA, P. MATYASIK, M. WYPYCH. *Generation of Labelled Transition Systems for Alvis Models Using Haskell Model Representation*, in "Proceedings of the 22nd International Workshop on Concurrency, Specification and Programming CS&P'2013 (Warsaw, Poland)", M. SZCZUKA, L. CZAJA, M. KACPRZAK (editors), September 2013, pp. 409–420

[67] M. TIMMER. , *Efficient Modelling, Generation and Analysis of Markov Automata*, University of Twente, September 2013

[68] J. TRETMANS. *Model Based Testing with Labelled Transition Systems*, in "Formal Methods and Testing, An Outcome of the FORTEST Network, Revised Selected Papers", R. M. HIERONS, J. P. BOWEN, M. HARMAN (editors), Lecture Notes in Computer Science, Springer Verlag, 2008, vol. 4949, pp. 1–38

[69] F. W. VAANDRAGER. *Active Learning of Extended Finite State Machines*, in "Proceedings of the 24th IFIP WG 6.1 International Conference on Testing Software and Systels ICTSS'2012 (Aalborg, Denmark)", Lecture Notes in Computer Science, Springer Verlag, November 2012, vol. 7641, pp. 5–7

[70] D. VEKRIS, C. DIMA. *Efficient Operational Semantics for EB3 for Verification of Temporal Properties*, in "Proceedings of the 5th International Conference on Fundamentals of Software Engineering (Tehran, Iran)", F. ARBAB, M. SIRJANI (editors), Lecture Notes in Computer Science, Springer Verlag, April 2013, vol. 8161, pp. 133–149

[71] A. WIJS, L. ENGELEN. *Efficient Property Preservation Checking of Model Refinements*, 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), Lecture Notes in Computer Science, Springer Verlag, March 2013, vol. 7795, pp. 565–579

[72] M. ZHANG. , *Scalably Verifiable Cache Coherence*, Duke University, 2013

[73] H. ZHAO, J. SUN, X. LIU. *A Model Checking Based Approach to Automatic Test Suite Generation for Testing Web Services and BPEL*, in "Proceedings of the Asia-Pacific Services Computing Conference APSCC'2012 (Guilin, China)", IEEE Computer Society Press, December 2012, pp. 61–69

[74] N. ZHAR, M. A. ALI, M. ELEULDJ, A. RAJI. *VIP DESIGN: Graphical Language for Image and Video Processing Embedded Systems Design*, in "Proceedings of the International Conference on Complex Systems ICCS'2012 (Agadir, Morocco)", IEEE Computer Society Press, November 2012, pp. 1–6

[75] N. ZHAR, M. A. ALI, M. ELEULDJ. *Toward a Graphical Tool for Image and Video Processing Embedded Systems Design*, in "Proceedings of the 2nd International Conference on Innovative Computing Technology INTECH'2012 (Casablanca, Morocco)", IEEE Computer Society Press, September 2012, pp. 158–163