IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2013

Project-Team VERIDIS

Modeling and Verification of Distributed Algorithms and Systems

IN COLLABORATION WITH: Laboratoire lorrain de recherche en informatique et ses applications (LORIA)

RESEARCH CENTER
Nancy - Grand Est

THEME
Proofs and Verification

Table of contents

Project-Team VERIDIS

Keywords: Formal Methods, Distributed System, Automated Theorem Proving, Interactive Theorem Proving, Model-checking

VeriDis is a joint research group of CNRS, Inria, Max-Planck-Institut für Informatik, and Université de Lorraine. It consists of members of the Mosel team at LORIA, Nancy, France, and members of the Automation of Logic group at Max-Planck-Institut für Informatik in Saarbrücken, Germany.

Creation of the Team: 2010 January 01*, updated into Project-Team:* 2012 July 01.

1. Members

Research Scientists

Stephan Merz [Team leader, Inria, Senior Researcher, HdR]
Thomas Sturm [Max-Planck-Institut für Informatik, Saarbrücken, Senior Researcher, HdR]
Uwe Waldmann [Max-Planck-Institut für Informatik, Saarbrücken, Senior Researcher]
Christoph Weidenbach [Team leader, Max-Planck-Institut für Informatik, Saarbrücken, Senior Researcher, HdR]

Faculty Members

Marie Duflot-Kremer [Univ. de Lorraine, Associate Professor]
Pascal Fontaine [Univ. de Lorraine, Associate Professor]
Dominique Méry [Univ. de Lorraine, Professor, HdR]

Engineer

Pablo Dobal [Inria]

PhD Students

Manamiary Andriamiarina [Univ. de Lorraine, since Oct 2010]
Noran Azmy [Univ. des Saarlandes, since Sep 2013]
Haniel Barbosa [Inria, CORDI-S, from Dec 2013]
Henri Debrat [Univ. de Lorraine, until Dec 2013]
Marek Košta [Univ. des Saarlandes, since Nov 2011]
Tianxiang Lu [Univ. des Saarlandes, until Oct 2013]
Hernán Vanzetto [Inria, CORDI-C, granted by Microsoft Research, since Nov 2010]

Post-Doctoral Fellow

Jingshu Chen [Inria, granted by Fondation d'entreprise EADS, from Sep 2013]

Visiting Scientists

Luciana Benotti [Univ. Nacional de Córdoba, Argentina, 08/2013]
Rodrigo Castaño [Univ. de Buenos Aires, Argentina, 10–12/2013]
David Déharbe [Associate Professor, on sabbatical from Univ. Federal do Rio Grande de Norte, Brazil, 08/2013–07/2014]
Raúl Fervari [Univ. Nacional de Córdoba, Argentina, 03/2013]
Guillaume Hoffmann [Univ. Nacional de Córdoba, Argentina, 06–10/2013]
Josef Widder [TU Wien, Austria, 10–11/2013]

Administrative Assistant

Sophie Drouot [Inria]

Others

Bhargav Bhatt [Inria, student intern, 05–07/2013]
Anisia Maria Magdalena Tudorescu [Inria, student intern, 03–05/2013]
Paula Chocrón [Inria, student intern, 10–12/2013, joint with Cassis team]

2. Overall Objectives

2.1. Introduction

The VeriDis project team includes members of the MOSEL team of LORIA, the computer science laboratory in Nancy, and members of the Automation of Logic Research Group at Max-Planck-Institut für Informatik (MPI-INF) in Saarbrücken. It is headed by Stephan Merz and Christoph Weidenbach. VeriDis was created in 2010 as a local team of Inria Nancy Grand-Est and has been an Inria project team since July 2012.

The objectives of VeriDis are to contribute to the advances in automated and interactive theorem proving and to make them available for the formal development of concurrent and distributed algorithms and systems, within the framework of mathematically precise and practically applicable development methods. We intend to assist algorithm and system designers carrying out formally proved developments, where proofs of relevant properties, as well as bugs, can be found with a high degree of automation.

Automated as well as interactive deduction techniques are already having substantial impact. In particular, they have been successfully applied to the verification and analysis of sequential programs, often in combination with static analysis and software model checking. Ideally, systems and their properties would be specified in high-level, expressive languages, errors in specifications would be discovered automatically, and finally, full verification could also be performed completely automatically. Due to the inherent complexity of the problem this cannot be achieved in general. However, we have observed important advances in automated and interactive theorem proving in recent years. We are particularly interested in the integration of different deduction techniques and tools, including the combination of relevant theories such as arithmetic in automated theorem proving. These advances suggest that a substantially higher degree of automation can be achieved in system verification over what is available in today's verification tools.

VeriDis proposes to exploit and further develop automation in system verification, and to apply its techniques within the context of concurrent and distributed algorithms, which are by now ubiquitous and whose verification is a big challenge. Concurrency problems are central to the development and verification of programs for multi- and many-core architectures, and distributed computation underlies the paradigms of grid and cloud computing. The potential of distributed systems for increased resilience to component failures makes them attractive in many contexts, but also makes formal verification important and challenging. We aim to move current research in this area on to a new level of productivity and quality. To give a concrete example: today the designer of a new distributed protocol may validate it using testing or model checking. Model checking will help finding bugs, but can only guarantee properties of a high-level model of the protocol, usually restricted to finite instances. Testing distributed systems and protocols is notoriously difficult because corner cases are hard to establish and reproduce. Also, many testing techniques require implementation, which is expensive and time-consuming, and errors are found only when they can no longer be fixed cheaply. The techniques that we develop aim at automatically proving significant properties of the protocol already at the design phase. Our methods will be applicable to designs and algorithms that are typical for components of operating systems, distributed services, and down to the (mobile) network systems industry.

2.2. Highlights of the Year

Uwe Waldmann received a LICS Test of Time Award for the paper "Set constraints are the monadic class" published at LICS 1993 together with Leo Bachmair and Harald Ganzinger. He also won the TFA category (typed first-order logic with arithmetic) of the CADE ATP System Competition 2013 using the prover SPASS+T.

Pascal Fontaine was the main organizer and program committee chair (with Christophe Ringeissen and Renate Schmidt) of FroCos 2013 in September in Nancy.

3. Research Program

3.1. Automated and interactive theorem proving

The VeriDis team unites experts in techniques and tools for interactive and automated verification, and specialists in methods and formalisms for the proved development of concurrent and distributed systems and algorithms. Our common objective is to advance the state of the art of combining interactive with automated methods resulting in powerful tools for the (semi-)automatic verification of distributed systems and protocols. Our techniques and tools will support methods for the formal development of trustworthy distributed systems that are grounded in mathematically precise semantics and that scale to algorithms relevant for practical applications.

The VeriDis members from Saarbrücken are developing SPASS [7], one of the leading automated theorem provers for first-order logic based on the superposition calculus [31]. Recent extensions to the system include the integration of dedicated reasoning procedures for specific theories, such as linear arithmetic [50], [29], that are ubiquitous in the verification of systems and algorithms. The group also studies general frameworks for the combination of theories such as the locality principle [51] and automated reasoning mechanisms these induce.

The VeriDis members from Nancy develop veriT [1], an SMT (Satisfiability Modulo Theories [33]) solver that combines decision procedures for different fragments of first-order logic and that integrates an automatic theorem prover for full first-order logic. The veriT solver is designed to produce detailed proofs; this makes it particularly suitable as a component of a robust cooperation of deduction tools.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction. Members of VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we participate in a project at the joint MSR-Inria Centre in Saclay on the development of methods and tools for the formal proof of TLA$^+$ [44] specifications. Our prover relies on a declarative proof language and includes several automatic backends [3].

3.2. Methodology of proved system development

Powerful theorem provers are not a panacea for system verification: they support sound methodologies for modeling and verifying systems. In this respect, members of VeriDis have gained expertise and recognition in making contributions to formal methods for concurrent and distributed algorithms and systems [2], [6], and in applying them to concrete use cases. In particular, the concept of *refinement* [28], [32], [48] in state-based modeling formalisms is central to our approach. Its basic idea is to derive an algorithm or implementation by providing a series of models, starting from a high-level description that precisely states the problem, and gradually adding details in intermediate models. An important goal in designing such methods is to reduce the number of generated proof obligations and/or support their proof by automatic tools. This requires taking into account specific characteristics of certain classes of systems and tailoring the model to concrete computational models. Our research in this area is supported by carrying out case studies for academic and industrial developments. This activity benefits from and influences the development of our proof tools.

Our vision for the integration of our expertise can be resumed as follows. Based on our experience and related work on specification languages, logical frameworks, and automatic theorem proving tools, we develop an approach that is suited for specification, interactive theorem proving, and for eventual automated analysis and verification, possibly through appropriate translation methods. While specifications are developed by users inside our framework, they are analyzed for errors by our SMT based verification tools. Eventually, properties are proved by a combination of interactive and automatic theorem proving tools.

Today, the formal verification of a new algorithm is typically the subject of a PhD thesis, if it is addressed at all. This situation is not sustainable given the move towards more and more parallelism in mainstream systems: algorithm developers and system designers must be able to productively use verification tools for validating their algorithms and implementations. On a high level, the goal of VeriDis is to make formal verification standard practice for the development of distributed algorithms and systems, just as symbolic model checking has become commonplace in the development of embedded systems and as security analysis for cryptographic protocols is becoming standard practice today. Although the fundamental problems in distributed programming, such as mutual exclusion, leader election, group membership or consensus, are well-known, they pose new challenges in the context of modern system paradigms, including ad-hoc and overlay networks or peer-to-peer systems.

4. Application Domains

4.1. Application Domains

Our work focuses on the formal modeling and verification of distributed algorithms and protocols. These are or will be found at all levels of computing infrastructure, from many-core processors and systems-on-chip to wide-area networks. We are particularly interested in novel paradigms, for example ad-hoc networks that underly mobile and low-power computing or overlay networks and peer-to-peer networking that provide services for telecommunication or cloud computing services. Distributed protocols underly computing infrastructure that must be highly available and mostly invisible to the end user, therefore correctness is important. One should note that standard problems of distributed computing such as consensus, group membership or leader election have to be reformulated for the dynamic context of these modern systems. We are not ourselves experts in the design of distributed algorithms, but work together with domain experts on the modeling and verification of these protocols. These cooperations help us focus on concrete algorithms and ensure that our work is relevant to the distributed algorithm community.

Formal verification techniques that we study can contribute to certify the correctness of systems. In particular, they help assert under which assumptions an algorithm or system functions as required. For example, the highest levels of the Common Criteria for Information Technology Security Evaluation require code analysis, based on mathematically precise foundations. While initially the requirements of certified development have mostly been restricted to safety-critical systems, they are becoming more and more common due to the cost associated with malfunctioning system components and software. For example, we are working on modeling and verifying medical devices that require closed-loop models of both the system and its environment.

5. Software and Platforms

5.1. The veriT solver

Participants: David Déharbe, Pablo Dobal, Haniel Barbosa, Pascal Fontaine [correspondent].

The veriT solver is an SMT (Satisfiability Modulo Theories) solver developed in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brazil. The solver can handle large quantifier-free formulas containing uninterpreted predicates and functions, and arithmetic over integers and reals. It features a very efficient decision procedure for difference logic, as well as a simplex-based reasoner for full linear arithmetic. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about concepts involving sets, relations, etc. The prover can produce an explicit proof trace when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, a regression platform using Inria's grid infrastructure is used; it allows us to extensively test the solver on thousands of benchmarks in a few minutes. The veriT solver is available as open source under the BSD license at the veriT Web site.

Efforts in 2013 have been focused on efficiency, and more specifically on arithmetic. A preliminary prototype integrating the solver Redlog for non-linear arithmetic has been stabilized. First results are encouraging; this prepares the ground for the starting ANR project SMArT (Satisfiability Modulo Arithmetic Theories), involving both sites of the VeriDis team (veriT being developed in Nancy and Redlog being designed in Saarbrücken), as well as Systerel as an industrial partner.

In late 2013, Haniel Barbosa joined the team as a PhD student. He will work on theoretical and practical aspects of handling quantifiers in SMT frameworks, which is currently an important challenge for SMT, and he will implement his techniques in veriT.

We target applications where validation of formulas is crucial, such as the validation of TLA$^+$ and B specifications, and work together with the developers of the respective verification platforms to make veriT even more useful in practice. The solver is available as a plugin for the Rodin platform for discharging proof obligations generated in Event-B [39]; on a large repository of industrial and academic cases, this SMT-based plugin decreased by 75% the number of proof obligations requiring human interactions, compared to the original B prover.

5.2. The TLA+ proof system

Participants: Bhargav Bhatt, Stephan Merz [correspondent], Hernán Vanzetto.

TLAPS, the TLA$^+$ proof system, is a platform for developing and mechanically verifying proofs about TLA$^+$ specifications. It is developed at the Joint MSR-Inria Centre. The TLA$^+$ proof language is hierarchical and explicit. TLAPS consists of a *proof manager* that interprets the proof language and generates a collection of proof obligations that are sent to *backend verifiers* that include theorem provers, proof assistants, SMT solvers, and decision procedures.

The current version 1.2.1 of TLAPS was released in September 2013, it is distributed under a BSD-like license at http://tla.msr-inria.inria.fr/tlaps/content/Home.html. The prover currently handles the non-temporal part of TLA$^+$ and can be used to prove safety, but not liveness properties. Its backends include a tableau prover for first-order logic, an encoding of TLA$^+$ in the proof assistant Isabelle, and a backend for interfacing with SMT solvers. The SMT backend, developed in Nancy, has been further improved in 2013 and is now considered by users as the most useful backend prover for system verification. During his internship in the summer of 2013, Bhargav Bhatt helped design and implement a standard library of TLA$^+$ theorems about functions, sequences, and finite sets that is now part of the TLAPS distribution. Development of support for temporal reasoning in TLAPS has started in late 2013.

6. New Results

6.1. Automated and Interactive Theorem Proving

6.1.1. Using symmetries in SMT

Participants: David Déharbe, Pascal Fontaine, Stephan Merz.

Joint work with Carlos Areces, Raúl Fervari, Guillaume Hoffmann, and Ezequiel Orbe at Universidad Nacional de Córdoba (see also section 8.2).

Methods exploiting problem symmetries have been very successful in several areas including constraint programming and SAT solving. We proposed similar techniques for enhancing the performance of SMT-solvers by detecting symmetries in the input formulas and using them to prune the search space of the SMT algorithm. These techniques are based on the concept of (syntactic) invariance by permutation of symbols. In 2011, we presented a technique restricted to constants but which exhibited impressive results for some categories of formulas [4]; this technique was quickly implemented in major SMT solvers, including CVC4 and Z3.

In 2013, we proposed, together with our colleagues at the University of Córdoba, Argentina, a more general approach to detect symmetries in an SMT context. These techniques are based on graph isomorphisms, and the Schreier-Sims algorithm for improving the presentation of the symmetries. This work was published at the SMT workshop 2013 [21].

6.1.2. Computing minimal models (prime implicants)

Participants: David Déharbe, Pascal Fontaine.

Joint work with Daniel Le Berre and Bertrand Mazure from the CRIL laboratory in Lens, France.

Model checking and counter-example guided abstraction refinement are examples of applications of SAT solving that require the production of models for satisfiable formulas. Instead of giving a truth value to every variable, it is usually preferable to provide an implicant, i.e. a partial assignment of the variables such that every full extension is a model for the formula. An implicant is *prime* if every assignment is necessary. Since prime implicants contain no literal irrelevant for the satisfiability of the formula, they are considered as highly refined information.

In 2013, we proposed a novel algorithm that uses data structures found in modern CDCL SAT solvers for efficiently computing prime implicants starting from an existing model. The original aspects are (1) the algorithm is based on watched literals and a form of propagation of required literals, adapted to CDCL solvers, (2) the algorithm works not only on clauses, but also on generalized constraints, and (3) for clauses (and more generally, for cardinality constraints) the complexity of the algorithm is linear in the size of the constraints. We implemented and evaluated the algorithm with the Sat4j library. This work gave rise to a publication at the FMCAD 2013 international conference [13].

6.1.3. Encoding TLA+ proof obligations for SMT solvers

Participants: Stephan Merz, Hernán Vanzetto.

The TLA$^+$ proof system TLAPS (see section 5.2) is being developed within a project at the MSR-Inria Joint Centre to which we contribute. Typical proof obligations that arise during the verification of TLA$^+$ specifications mix reasoning about sets, functions, arithmetic, tuples, and records. In previous work [47], we have developed translations from TLA$^+$ set theory to SMT-Lib, the standard input language of SMT solvers. The main challenge has been to design a sound translation from untyped TLA$^+$ to the multi-sorted first-order logic that underlies SMT-Lib. Our solution is based on an incomplete type inference based on "typing hypotheses" present in TLA$^+$ proof obligations. When type inference fails, we fall back to an "untyped" encoding where interpreted sorts such as integers are injected into a designated sort of TLA$^+$ values, and proof obligations corresponding to well-sortedness conditions must be discharged during the proof.

In 2013, we have stabilized and extended the type inference, based on a more expressive type system that includes dependent types, predicate types, and subtyping. The new type system is able to solve many more typing conditions during the translation of proof obligations and thus improves both the scope and the efficiency of the SMT backend. It has been implemented as part of the SMT backend of TLAPS, and an article describing the type system has been submitted. A full description will appear in the PhD thesis of Hernán Vanzetto, expected to be defended in early 2014.

6.1.4. Formalization of stuttering invariance in temporal logic

Participant: Stephan Merz.

Extending our previous formalization in the interactive proof assistant Isabelle/HOL of the concept of stuttering invariance, we formally proved that a property expressible in propositional temporal logic is stuttering invariant if and only if it is equivalent to a formula using only the *until* temporal operator (and in particular not the *next-time* operator). The formalization follows the proof in the classical paper by Peled and Wilke [49]. It allowed us to uncover and correct an error in the proof that had previously not been known. The corresponding extended version of the Isabelle proof development has been accepted at the Archive of Formal Proofs.

6.1.5. Superposition modulo theories

Participants: Noran Azmy, Christoph Weidenbach.

We are currently in a transition phase moving SPASS from a first-order logic prover to a first-order logic prover over theories SPASS(T), in particular arithmetic. Our experience in combining SPASS with interactive verification systems such as TLAPS or Isabelle shows that this is a mandatory step in improving automation [46], [34]. Meanwhile we have built the theoretical foundations [41], [40], [43] for combining superposition with theories which we now turn into algorithmic solutions. This makes an overall reimplementation of SPASS necessary. As a first step we reimplemented and improved our clause normal form transformation [11].

In particular, we want to support integer theories and modulo reasoning [15], as it is often used in distributed algorithms [46]. We have built first implementations of arithmetic modules which we want to combine in 2014 to a first version of SPASS(T).

6.1.6. Presburger Arithmetic in Compiler Optimization

Participants: Marek Košta, Thomas Sturm.

One of our focuses in 2013 was the application of SMT-solvers in new and different problem areas. We started a fruitful cooperation with the Compiler Lab at the Saarland University, Germany on compilation of data-parallel languages.

Data-parallel languages like OpenCL and CUDA are an important means to exploit the parallel computational capabilities of today's computing devices. However, the historical development of data-parallel languages stemming from GPUs plays a crucial role when compiling them for a SIMD (Single Instruction Multiple Data) CPU: on the CPU, one has to emulate dynamic features that on GPU are implemented in the hardware. This difference gives rise to several problems that have to be dealt with during the compilation process.

Our work [15] considers compilation of OpenCL programs for CPUs with SIMD instruction sets. It turns out that SMT-solvers can be used to generate more efficient CPU code. The lack of some dynamic features on CPU implies that one wants to statically decide whether or not certain memory operations access consecutive addresses. Our approach formalizes the notion of consecutivity and algorithmically reduces the static decision to satisfiability problems in Presburger Arithmetic. This is where SMT-solvers come into play. To make an application of an off-the-shelf SMT solver feasible, a preprocessing technique on the SMT problems was introduced. Combining three different systems (computer algebra system REDLOG, SMT-solver Z3, and an OpenCL driver developed in the Compiler Lab), a proof-of-concept system based on our approach was developed. The system generated more efficient code than any other state-of-the-art OpenCL compiler.

Further development is needed to turn the proof-of-concept system mentioned above into one integrated software system. To achieve this, the redundant combination of three heterogeneous systems needs to be replaced by a coherent library offering the same functionality. The work [23] presents the development of such a novel library. The library provides functions to fully automatize the approach proposed in the previous work. It is capable of parallel computations by means of threads and processes and uses an SMT-solver library to carry out the needed computations. To create the final system, the integration of the library with the OpenCL driver needs to be done. This final step is left for future work.

6.1.7. Non-Linear SMT-Solving

Participants: Marek Košta, Thomas Sturm.

In [42] de Moura and Jovanović give a novel satisfiability procedure for the theory of the reals. The procedure uses DPLL-style techniques to search for a satisfying assignment. In case of a conflict, cylindrical algebraic decomposition (CAD) [38] is used to guide the search away from the conflicting state: on the basis of one conflicting point, the procedure learns to avoid in the future an entire CAD cell containing the point. The function realizing this learning is the crucial ingredient that makes the DPLL-style search possible at all. Unfortunately, it is the main computational bottleneck of the whole procedure.

The work of Brown [35] develops a more efficient learning function for the case when the cell to-be learned is full-dimensional. In collaboration with Prof. Brown (United States Naval Academy, USA), we extend this to the general case. While restricting to one cell is quite straightforward for the base and lifting phases of a CAD algorithm, our approach is able to optimize the projection phase as well. This requires a thorough analysis of available geometric infomation and properties of the involved projection operator. Our cell construction algorithm is able to produce bigger cells and it is faster than the approach used in [42]. Both of these are benefits, because a bigger cell means a better generalization of the conflicting assignment. Prototypical implementation of our cell construction algorithm gives very promising results on various kinds of problems. Its elaborate implementation and integration with an DPLL engine within the computer algebra system REDLOG is left for future work. A publication has been submitted to the Journal of Symbolic Computation.

6.1.8. Towards Tropical Decision for NLA

Participant: Thomas Sturm.

Inspired by problems related to stability analysis of chemical reaction networks we have developed an incomplete decision procedure for satisfiability in nonlinear real arithmetic. A first implemented version focuses on specific situations where all variables are known to be stricly positive, which naturally occurs in many scientific contexts. Furthermore, only one single equation is considered. The principal *tropical* approach is, after reducing the problem to finding a point with positive value for f in the considered equation $f = 0$, to consider instead of f only the exponent tuples of the contained summands as points in $\mathbb{Z}^n$. On that basis dominating summands can be identified using LP techniques.

In our particular application discussed in [14], we were able to solve problems, which are intractable even by numerical methods: Typical input equations had around 6000 summands and up to seven variables of degrees between 4 and 9. The methods failed in only 3 percent of the 496 considered input problems.

We are currently generalizing the approach to the general case where variables can have arbitrary values. Furthermore, as it is well known that every existential decision problems over the reals can be equi-satisfiably encoded into one equation, we are aiming at a corresponding general procedure as a long-term research goal.

6.1.9. Hierarchical superposition for arithmetic

Participant: Uwe Waldmann.

Many applications of automated deduction require reasoning in first-order logic modulo background theories, in particular some form of integer arithmetic. A major unsolved research challenge is to design theorem provers that are "reasonably complete" even in the presence of free function symbols ranging into a background theory sort. The hierarchic superposition calculus of Bachmair, Ganzinger, and Waldmann already supports such symbols, but not optimally. We have introduced a novel form of clause abstraction, a core component in the hierarchic superposition calculus for transforming clauses into a form needed for internal operation. We have also demonstrated that hierarchic superposition is refutationally complete for linear integer or rational arithmetic, even if one considers the standard model semantics rather than the first-order semantics, provided that all background-sorted terms in the input are either ground or variables (variables with integer offsets can be permitted in certain positions).

6.2. Proved development of algorithms and systems

6.2.1. Incremental development of distributed algorithms

Participants: Dominique Méry, Manamiary Andriamiarina.

Joint work with Mohammed Mosbah and Mohammed Tounsi from the LABRI laboratory in Bordeaux, France.

The development of distributed algorithms and, more generally, of distributed systems, is a complex, delicate, and challenging process. The approach based on refinement helps to gain formality by using a proof assistant, and proposes to apply a design methodology that starts from the most abstract model and leads, in an incremental way, to the most concrete model, for producing a distributed solution. Our work helps formalizing pre-existing algorithms, developing new algorithms, as well as developing models for distributed systems.

Our research was initially supported by the ANR project RIMEL (see http://rimel.loria.fr). More concretely, we aim at an integration of the correct-by-construction refinement-based approach into the *local computation* programming model. The team of LABRI develops an environment called VISIDIA (http://visidia.labri.fr) that provides a toolset for developing distributed algorithms expressed as a set of rewriting rules of graph structures. The simulation of rewriting rules is based on synchronization algorithms, and we have developed these algorithms by refinement [20].

In particular, we show how state-based models can be developed for specific problems and how they can be simply reused by controlling the composition of state-based models through the refinement relationship. Traditionally, distributed algorithms are supposed to run on a fixed network, whereas we consider a network with a changing topology.

The contribution is related to the development of proof-based patterns providing effective help to the developer of formal models of applications [10]. Our patterns simplify the development of distributed systems using refinement and temporal logic. Moreover, we have especially evaluated the extension of the scope of Event B by proposing a technique for integrating fairness in the development of distributed algorithms [17].

6.2.2. Modeling Medical Devices

Participant: Dominique Méry.

Formal modelling techniques and tools have attained sufficient maturity for formalizing highly critical systems in view of improving their quality and reliability, and the development of such methods has attracted the interest of industrial partners and academic research institutions. Building high quality and zero-defect medical software-based devices is a particular domain where formal modelling techniques can be applied effectively. Medical devices are very prone to showing unexpected system behaviour in operation when traditional methods are used for system testing. Device-related problems have been responsible for a large number of serious injuries. Officials of the US Food and Drug Administration (FDA) found that many deaths and injuries related to these devices are caused by flaws in product design and engineering. Cardiac pacemakers and implantable cardioverter-defibrillators (ICDs) are among the most critical medical devices and require closed-loop modelling (integrated system and environment modelling) for verification purposes before obtaining a certificate from the certification bodies.

Clinical guidelines systematically assist practitioners in providing appropriate health care in specific clinical circumstances. Today, a significant number of guidelines and protocols are lacking in quality. Indeed, ambiguity and incompleteness are likely anomalies in medical practice. The analysis of guidelines using formal methods is a promising approach for improving them.

In [9], we propose a refinement-based methodology for complex medical systems design, which possesses all the required key features. A refinement-based combined approach of formal verification, model validation using a model-checker and refinement chart is proposed in this methodology for designing a high-confidence medical device. Furthermore, we show the effectiveness of this methodology for the design of a cardiac pacemaker system.

Inappropriate mode transitions can be a common cause of mishaps in complex health-care systems. In [19], we present an approach for formalizing and reasoning about optimal mode transition in a health-care system that uses several operating modes in various operating states. Modes are formalized and their relation to a state-based formalism is established through a refinement approach. The efficiency of this approach is presented by formalizing an ideal operating mode transition of a cardiac pacemaker case study. An incremental approach is used to develop the system and its detailed design is verified through a series of refinements. In this way, we show how to improve system structuring, elicitation of system assumptions and expected functionality, as well as requirement traceability using modes in state-based modeling. Models are expressed in the Event B [25] modeling language, and they are validated by the model checker ProB.

Finally, in a joint work with colleagues of the CRAN laboratory in Nancy, we have completed a joint project with Airbus on the integration of physiological features in the development of systems like maintenance systems.

6.2.3. Analysis of real-time Java programs

Participants: Jingshu Chen, Marie Duflot-Kremer, Pascal Fontaine, Stephan Merz.

Joint work with Nadezhda Baklanova, Jan-Georg Smaus, Wilmer Ricciotti, and Martin Strecker at IRIT Toulouse, France, funded by EADS Foundation (see also section 7.1).

We investigate techniques for the formal verification of programs written in a dialect of Java that includes real-time annotations. Inspired by Safety-Critical Java [36], our partners in Toulouse developed a formal semantics for that dialect in Isabelle/HOL. In joint work, we have designed translations of programs to respectively timed automata and to SMT-Lib for analysis with the Uppaal model checker and with SMT solvers. We are evaluating the features and the scalability of the two approaches, and also plan to formally prove the soundness of the translations based on the semantics formalized in Isabelle.

6.2.4. Fundamentals of Network Calculus in Isabelle/HOL

Participant: Stephan Merz.

Joint work with Marc Boyer from ONERA (Toulouse, France) and Loïc Fejoz, Etienne Mabille and Nicolas Navet from RealTime at Work (RTaW, Nancy).

Network Calculus [45] is a well-established theory for the design and analysis of embedded networks. Based on the $(\min, +)$ dioid, it allows a network designer to compute upper bounds for delay and buffer sizes in networks. The theory is supported by several commercial and open-source tools and has been used in major industrial applications, such as the design and certification of the Airbus A380 AFDX backbone. Nevertheless, it is difficult for certification authorities to assess the correctness of the computations carried out by the tools supporting Network Calculus, and we propose the use of *result certification* techniques for increasing the confidence in the Network Calculus toolchain. We have formalized parts of the theory underlying Network Calculus in the proof assistant Isabelle/HOL. We have also developed a prototype analyzer that outputs traces of its computations so that they can be certified using Isabelle. Our work has been published at the conferences EUCASS and ITP [16], [24], and we have submitted a project proposal to ANR together with ONERA, RTaW, Kalray, Eurocopter, and Astrium. Unfortunately, the project was not granted, and future work on this promising subject is on hold.

6.2.5. Modeling and verifying the Pastry routing protocol

Participants: Tianxiang Lu, Stephan Merz, Christoph Weidenbach.

As a significant case study for the techniques that we are developing within VeriDis, we are modeling and verifying the routing protocol of the Pastry algorithm [37] for maintaining a distributed hash table in a peer-to-peer network. As part of his PhD work, Tianxiang Lu developed a TLA$^+$ model of the Pastry routing protocol, and has uncovered several problems in the existing presentations of the protocol in the literature that could lead to network partitioning.

He proposed a novel variant of the protocol and proved its correctness under the strong assumption that no nodes leave the network, using TLAPS (see section 5.2). He also demonstrated that the protocol could not work if arbitrary nodes are allowed to leave; it is not clear at this point under what reasonable assumptions the protocol can be made to work. The correctness proofs contain almost 15000 interactions and constitutes the largest case study carried out so far using TLAPS. Tianxiang Lu defended his thesis at the end of November 2013; a journal publication describing this work is in preparation.

6.2.6. Bounding message length in attacks against security protocols

Participant: Marie Duflot-Kremer.

Joint work with Myrto Arapinis from the University of Birmingham, UK.

Security protocols are short programs that describe communication between two or more parties in order to achieve security goals. Despite the apparent simplicity of such protocols, their verification is a difficult problem and has been shown to be undecidable in general. This undecidability comes from the fact that the set of executions to be considered is of infinite depth (an infinite number of protocol sessions can be run) and infinitely branching (the intruder can generate an unbounded number of distinct messages). Several attempts have been made to tackle each of these sources of undecidability. We have shown [30] that, under a syntactic and reasonable condition of “well-formedness” on the protocol, we can get rid of the infinitely branching part. Following this conference publication, we have submitted a journal version of this result extending the set of security properties to which the result is applicable, in particular including authentication properties.

6.2.7. Evaluating and verifying probabilistic systems

Participant: Marie Duflot-Kremer.

Joint work with colleagues at ENS Cachan and University Paris Est Créteil.

Since its introduction in the 1980s, model checking has become a prominent technique for the verification of complex systems. The aim was to decide whether or not a system was fulfilling its specification. With the rise of probabilistic systems, new techniques have been designed to verify this new type of systems, and appropriate logics have been proposed to describe more subtle properties to be verified. However, some characteristics of such systems cannot fall in the field of model checking. The aim is thus not to tell wether a property is satisfied but how well the system performs with respect to a certain measure. We have designed a statistical tool for tackling both performance and verification issues. Following several conference talks, two journal papers have been written. The first one presents the approach in details with a few illustrative applications. The second one focuses on biological application, and more precisely the use of statistical model checking to detect and measure several indicators of oscillating biological systems.

7. Bilateral Contracts and Grants with Industry

7.1. Project funded by EADS Foundation

Participants: Jingshu Chen, Marie Duflot-Kremer, Pascal Fontaine, Stephan Merz.

This two-year project (2013/2014) funds our work on the analysis of real-time Java programs described in section 6.2, and in particular 12 months of the salary of Jingshu Chen as a post-doctoral researcher. It is complemented by funds granted by Région Lorraine.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. Inria Development Action VeriT

Participants: Pablo Dobal, Pascal Fontaine.

Inria funds this project (started in 2011) to support the development of the SMT solver veriT (see section 5.1), including added expressiveness, improved efficiency and code stability, and interfaces with tools that embed veriT as a backend solver. The project is coordinated by Pascal Fontaine and also includes Inria Rennes (Celtique) and Sophia Antipolis (Marelle). Federico Dobal has been hired in 2012 on a position funded by this project and has in particular contributed to improvements in the code of the solver as well as of the testing platform that allows us to detect bugs and the impact of changes on the performance of the tool. He has also contributed to the maintenance of the deltaSMT tool, which has been used by several other teams of SMT developers for debugging SMT solvers.

8.2. European Initiatives

8.2.1. FP7 project MEALS

Type: PEOPLE
Instrument: International Research Staff Exchange Scheme
Objective: Exchange of scientists between Europe and Argentina
Duration: October 2011 - September 2015
Coordinator: Holger Hermanns, Universität des Saarlandes (Germany)
Partner: Universidad de Buenos Aires, Universidad Nacional de Córdoba, Universidad Nacional de Rio Cuarto, Instituto Tecnológico Buenos Aires
Inria contact: Castuscia Palamidessi
Abstract: The MEALS project funds exchanges between scientists in Europe (Saarland University, RWTH Aachen, TU Dresden, Inria, Imperial College, Univ. of Leicester, TU Eindhoven); it is structured in five work packages (Quantitative Analysis of Concurrent Program Behaviour, Reasoning Tasks for Specification and Verification, Security and Information Flow Properties, Synthesis in Model-based Systems Engineering, Foundations for the Elaboration and Analysis of Requirements Specifications). Our team mainly cooperates with the group led by Carlos Areces in Córdoba, as well with Diego Garbervetsky in Buenos Aires, within work package 2. In 2013, the project funded visits by Luciana Benotti, Rodrigo Castaño, Raúl Fervari, and Guillaume Hoffmann.

8.2.2. Cooperation with TU Wien, Austria

Participants: Pascal Fontaine, Stephan Merz.

This project – from January 2012 to December 2013 – fosters bilateral cooperation with the team headed by Prof. Alexander Leitsch at TU Vienna. It focuses on aspects of proof production and proof compression in automated reasoning. It is headed by Bruno Woltzenlogel Paleo of TU Wien, who was formerly a post-doctoral researcher in VeriDis until March 2011, and Pascal Fontaine. The project is funded by the Amadeus Programme of the Partenariat Hubert Curien and the Österreichischer Austausch Dienst.

The project funded the traveling costs for the participants for four one-week workshops in Vienna and Nancy. In particular, the third workshop was affiliated to Tableaux 2013 and was open to the participants of Tableaux; it attracted around 40 participants. The final workshop of the project took place in November 2013 in Vienna.

The discussions involved many aspects on proofs and allowed to improve some aspects of proof production in SMT, as well as several proof handling tools (e.g. Skeptik), developed among others at TU Wien. The web page gives more information on this project.

8.2.3. Cooperation with NUI Maynooth, Ireland

Participant: Dominique Méry.

The project *Building Reliable Systems: Software Refinement meets Software Verification* is a one-year project funded by PHC Ulysses. The academic Irish partner is Dr Rosemary Monahan of NUI Maynooth. The verification of software requires the specification of preconditions and postconditions as well as other properties of the code. These properties are expressed as annotations providing a detailed understanding of how the software is implemented. In program verification, the annotation process is often done *a posteriori*, with verification tools used to check that annotations are sound according to the semantics of the program. Determining the correct annotations to provide a complete specification is difficult, especially when specifying invariant properties of the code. *A priori* techniques for developing correct software are based on the correct-by-construction paradigm. The refinement-based approach is such a technique, providing for the construction of a correct program through the step-by-step refinement of an initial high-level model of the software. In this way, the program specification is developed alongside the code, discharging the conditions that need to be proved. We focus on combining these two software engineering techniques, to benefit from the strengths of both. We have proposed a framework [18] for integrating a representation of the *a posteriori* paradigm, namely Spec#, and a representation of the *a priori* paradigm, namely Event B. This integration induces a methodology which bridges the gap between software modeling and program verification in the software development life cycle.

8.3. International Initiatives

8.3.1. Participation In International Programs

8.3.1.1. Cooperation with Universidade Federal do Rio Grande de Norte, Brazil

Participants: David Déharbe, Pablo Dobal, Pascal Fontaine, Stephan Merz.

VeriDis has a close working relationship with a team at Universidade Federal do Rio Grande de Norte (UFRN), Brazil, and more specifically with Prof. David Déharbe. Pascal Fontaine visited Natal in early 2013. The project is centered around the development and applications of the veriT solver (section 5.1), of which David Déharbe and Pascal Fontaine are the main developers. Our cooperation was also supported by the Inria-CNPq project SMT-SAVeS from 2010 throughout early 2013.

A new STIC AmSud project has been approved that will start in 2014 and involves a team at the University of Córboba in Argentina, the team at UFRN, and VeriDis. It is again centered on SMT, with a particular focus on quantifiers and modal logic [21].

8.4. International Research Visitors

8.4.1. Visits of International Scientists

David Déharbe from UFRN (Natal, Brazil) joined the VeriDis team in Nancy for a one-year sabbatical that started in August 2013.

Josef Widder from TU Vienna, Austria, spent 6 weeks in Nancy in October and November 2013 as an Inria invited researcher. Together with Stephan Merz, he worked on the formalization of parameterized model checking techniques for fault-tolerant distributed algorithms in a proof assistant.

Mike Poppleton from the University of Southampton and Hoang Thai Son from ETHZ spent a week in our group for developing techniques to integrate fairness in Event B models, on the basis of the work published at IFM 2013 [17].

8.4.1.1. Internships

Luis Esteban Campostrini

Subject: Formal Verification of Distributed Algorithms

Date: from May until October, 2013

Institution: Universidad National de Rosario (Argentina)

Joint supervision with Martin Quinson (AlGorille team)

Anisia Maria Magdalena Tudorescu

Subject: Integrating SMT solvers into Spike

Date: from March 2013 until May 2013

Institution: West Timisoara University (Romania)

Joint supervision with Christophe Ringeissen (Cassis team) and Sorin Stratulat (Pareo team)

Paula Chocrón

Subject: Non-disjoint combination for SMT solvers: sharing a fragment of arithmetic

Date: from September 2013 until December 2013

Institution: University of Buenos Aires (Argentina)

Joint supervision with Christophe Ringeissen (Cassis team)

9. Dissemination

9.1. Scientific Animation

- Pascal Fontaine co-chaired the International Conference on Frontiers of Combining Systems (FroCoS 2013). He served on the program committee of the workshops PxTP 2013, SMT 2013, and the International Conference on Computer Aided Deduction (CADE 2013). He is an elected member of the SMT Steering Committee, and one of three SMT-LIB managers.
- Dominique Méry is
  - a member of the IFIP Working Group 1.3 on *Foundations of System Specification*,
  - head of the Doctoral School IAEM Lorraine for the University of Lorraine,
  - head of the Formal Methods department of the LORIA laboratory,
  - an expert for the French Ministry of Education (DS9),
  - an expert for the French Agence Nationale de la Recherche (ANR) and AERES.
  - He served on the program committees of FHIES, FM, ICECCS, ICFEM, iFM, and FACS.
- The academic duties of Stephan Merz in 2013 included:
  - member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*,
  - Inria representative in the Scientific Directorate of the International Computer Science Meeting Center in Dagstuhl,
  - delegate for the organization of conferences at Inria Nancy Grand-Est,
  - co-head of the PhD committee for computer science in Lorraine,
  - member of the program committees of iFM, Memocode, SAC, SBMF, and SEFM conferences, AFADL, AVoCS, Refinement, and SCSS workshops, member of the steering committee of AVoCS,
  - co-organizer of the VTSA summer school between Nancy, Saarbrücken, Luxembourg, and Liège,
  - president of the hiring committee for a professorship at Télécom Nancy and member of the hiring committee for professors at Université de l'Artois in Lens,
  - expert for the French Agence Nationale de la Recherche (ANR), German DFG, and Canadian NSERC.
- Thomas Sturm is a member of the Selection Committees for MSc and PhD students of the International Max-Planck Research School for Computer Science.
- Christoph Weidenbach is:
  - editor of JAR,
  - trustee of CADE Inc (elected 2009, reelected 2012),
  - member of the Appointment Decision Panel of FBK, Trento,
  - member of the Selection Committee of the Saarbruecken Graduate School in Computer Science,
  - member of Steering Committee *Bundeswettbewerb Informatik*,
  - co-organizer of the VTSA summer school between Nancy, Saarbrücken, Luxembourg, and Liège.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

The university employees of VeriDis have significant teaching obligations. We indicate the graduate courses they have been teaching this year.

- Dominique Méry gave courses in the Master program in Nancy on: formal system engineering, modeling and verification of systems, theoretical computer science, development of software systems, distributed algorithms.
- Marie Duflot-Kremer and Stephan Merz taught a course on algorithmic verification in the Master program in Nancy.
- Uwe Waldmann taught a course on Automated Reasoning at Saarland University.
- Christoph Weidenbach gave a course on Automated Reasoning II and lectured within the series "Perspektiven der Informatik" at Saarland University.

9.2.2. Supervision

- PhD: Henri Debrat, Certification formelle de la correction d'algorithmes de Consensus, Université de Lorraine. Supervised by Bernadette Charron-Bost and Stephan Merz, defended on December 6, 2013.
- PhD: Tianxiang Lu, Formal Verification of the Pastry Protocol, Université de Lorraine and Universität des Saarlandes. Supervised by Stephan Merz and Christoph Weidenbach, defended on November 27, 2013.
- PhD in progress: Manamiary Andriamiarina, Refinement Techniques for Distributed Algorithms, Université de Lorraine. Supervised by Dominique Méry, since 10/2010.
- PhD in progress: Noran Azmy, On the Automation of Proofs in TLAPS, Saarland University. Supervised by Christoph Weidenbach, since 11/2012.
- PhD in progress: Haniel Barbosa, Refutational Completeness in Satisfiability Modulo Theories, Université de Lorraine. Supervised by Pascal Fontaine and Stephan Merz, since 12/2013.
- PhD in progress: Marek Košta, Computational Logic, Universität des Saarlandes. Supervised by Thomas Sturm, since 11/2011.
- PhD in progress: Hernán Vanzetto, SMT Techniques for TLA$^+$ Proof Obligations, Université de Lorraine. Supervised by Kaustuv Chaudhuri and Stephan Merz, since 10/2010.

9.2.3. Juries

Stephan Merz wrote reports on the following PhD theses:

- Pierre-Emmanuel Cornilleau: *Certification of Static Analysis in Many-Sorted First-Order Logic*, ENS Cachan-Bretagne;
- Mélanie Jacquel: *Automatisation des preuves pour la vérification des règles de l'Atelier B*, CNAM Paris;
- Chantal Keller: *A Matter of Trust: Skeptical Communication Between Coq and External Provers*, Ecole Polytechnique;
- Yan Zhang: *Semi-Automatic Controller Design in a Java-like Language*, Université Paris 6.

He also was a member of the PhD committees of Dorin Maxim and Faqing Yang in Nancy.

Thomas Sturm was a member of the PhD committee of Evgeny Kruglov in Saarbrücken.

9.3. Popularization

Marie Duflot-Kremer, Pascal Fontaine, and Stephan Merz presented some of the subjects and techniques that underly formal verification of protocols and algorithms at events like "Fête de la Science". Using wooden puzzles, Sudoku sheets or boxes with locks, they explained how real-life problems can be represented in logical form and then solved using automated tools based on formal logic.

Marie Duflot-Kremer presented exercise sessions for high school students on "conducting a police investigation using databases" and "discovering Turing machines with Lego bricks". She is also a member of the steering committee preparing an itinerant exposition intended for explaining computer science to high-school students.

Thomas Sturm, Uwe Waldmann, and Christoph Weidenbach are involved in the “Computer Science Research Days” which take place every year. Gifted students from all over Germany can actively participate in current research themes within the Max Planck Institute for Informatics, the Computer Science Department of Saarland University and the German Research Center for Artificial Intelligence. The goal is to fill young people with enthusiasm for the subject of computer science as well as to discover and support the development of new talent.

10. Bibliography

Major publications by the team in recent years

[1] T. BOUTON, D. C. B. DE OLIVEIRA, D. DÉHARBE, P. FONTAINE. *veriT: an open, trustable and efficient SMT-solver*, in "Proc. Conference on Automated Deduction (CADE)", Montreal, Canada, R. SCHMIDT (editor), Lecture Notes in Computer Science, Springer, 2009, vol. 5663, pp. 151-156

[2] D. CANSELL, D. MÉRY. *The Event-B Modelling Method: Concepts and Case Studies*, in "Logics of Specification Languages", Berlin-Heidelberg, D. BJØRNER, M. C. HENSON (editors), Monographs in Theoretical Computer Science, Springer, 2008, pp. 47–152

[3] D. COUSINEAU, D. DOLIGEZ, L. LAMPORT, S. MERZ, D. RICKETTS, H. VANZETTO. *TLA+ Proofs*, in "18th International Symposium On Formal Methods - FM 2012", Paris, France, D. GIANNAKOPOULOU, D. MÉRY (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7436, pp. 147-154

[4] D. DÉHARBE, P. FONTAINE, S. MERZ, B. WOLTZENLOGEL PALEO. *Exploiting Symmetry in SMT Problems*, in "23rd Intl. Conf. Automated Deduction (CADE 2011)", Wroclaw, Poland, N. BJØRNER, V. SOFRONIE-STOKKERMANS (editors), LNCS, Springer, 2011, vol. 6803, pp. 222-236

[5] F. KRÖGER, S. MERZ. , *Temporal Logic and State Systems*, Texts in Theoretical Computer Science., Springer, 2008, 436 p. , http://hal.inria.fr/inria-00274806/en/

[6] S. MERZ. *The Specification Language TLA$^+$*, in "Logics of Specification Languages", Berlin-Heidelberg, D. BJØRNER, M. C. HENSON (editors), Monographs in Theoretical Computer Science, Springer, 2008, pp. 401–451

[7] C. WEIDENBACH, D. DIMOVA, A. FIETZKE, M. SUDA, P. WISCHNEWSKI. *SPASS Version 3.5*, in "22nd International Conference on Automated Deduction (CADE-22)", Montreal, Canada, R. SCHMIDT (editor), LNAI, Springer, 2009, vol. 5663, pp. 140–145

Publications of the year

Articles in International Peer-Reviewed Journals

[8] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Revisiting Snapshot Algorithms by Refinement-based Techniques (Extended Version)*, in "Computer Science and Information Systems", 2014, http://hal.inria.fr/hal-00924525

[9] D. MÉRY, N. K. SINGH. *Formal Specification of Medical Systems by Proof-Based Refinement*, in "ACM Transactions in Embedded Computing Systems", January 2013, vol. 12, n^o 1, 15 p. [*DOI :* 10.1145/2406336.2406351], http://hal.inria.fr/inria-00637756

International Conferences with Proceedings

[10] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Integrating Proved State-Based Models for Constructing Correct Distributed Algorithms*, in "iFM - 10th International Conference on integrated Formal Methods - 2013", Turku, Finland, June 2013, http://hal.inria.fr/hal-00819256

[11] N. AZMY, C. WEIDENBACH. *Computing Tiny Clause Normal Forms*, in "24th International Conference on Automated Deduction (CADE-24)", Lake Placid, NY, United States, M.-P. BONACINA (editor), Lecture Notes in Computer Science, Springer, 2013, vol. 7898, pp. 109-125 [*DOI :* 10.1007/978-3-642-38574-2_7], http://hal.inria.fr/hal-00931893

[12] P. BAUMGARTNER, U. WALDMANN. *Hierarchic Superposition With Weak Abstraction*, in "24th International Conference on Automated Deduction (CADE-24)", Lake Placid, NY, United States, M. P. BONACINA (editor), Lecture Notes in Computer Science, Springer, 2013, vol. 7898, pp. 39-57 [*DOI :* 10.1007/978-3-642-38574-2_3], http://hal.inria.fr/hal-00931919

[13] D. DÉHARBE, P. FONTAINE, D. LE BERRE, B. MAZURE. *Computing prime implicant*, in "FMCAD - Formal Methods in Computer-Aided Design 2013", Portland, United States, IEEE, October 2013, pp. 46-52, http://hal.inria.fr/hal-00910363

[14] H. ERRAMI, M. EISWIRTH, D. GRIGORIEV, W. SEILER, T. STURM, A. WEBER. *Efficient Methods to Compute Hopf Bifurcations in Chemical Reaction Networks Using Reaction Coordinates*, in "Computer Algebra in Scientific Computing", Berlin, Germany, V. P. GERDT, W. KOEPF, E. W. MAYR, E. V. VOROZHTSOV (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8136, pp. 88-99 [*DOI :* 10.1007/978-3-319-02297-0_7], http://hal.inria.fr/hal-00931946

[15] R. KARRENBERG, M. KOSTA, T. STURM. *Presburger Arithmetic in Memory Access Optimization for Data-Parallel Languages*, in "9th International Conference Frontiers of Combining Systems (FroCos 2013)", Nancy, France, P. FONTAINE, C. RINGEISSEN, R. SCHMIDT (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8152, pp. 56-70 [*DOI :* 10.1007/978-3-642-40885-4_5], http://hal.inria.fr/hal-00931954

[16] E. MABILLE, M. BOYER, L. FÉJOZ, S. MERZ. *Towards Certifying Network Calculus*, in "ITP - 4th International Conference on Interactive Theorem Proving", Rennes, France, S. BLAZY, C. PAULIN-MOHRING, D. PICHARDIE (editors), Lecture Notes in Computer Science, Springer, July 2013, vol. 7998, pp. 484-489 [*DOI :* 10.1007/978-3-642-39634-2_37], http://hal.inria.fr/hal-00904796

[17] D. MÉRY, M. POPPLETON. *Formal Modelling and Verification of Population Protocols*, in "iFM - 10th International Conference on integrated Formal Methods - 2013", Turku, Finland, E. B. JOHNSEN, L. PETRE (editors), LNCS, Springer, June 2013, http://hal.inria.fr/hal-00813033

[18] D. MÉRY, M. ROSEMARY. *Transforming EVENT B Models into Verified C# Implementations*, in "VPT 2013 - First International Workshop on Verification and Program Transformation", Saint Petersburg, Russian Federation, A. LISITSA, A. NEMYTYKH (editors), EPIC, Alexei Lisitsa and Andrei Nemytykh, July 2013, vol. 16, pp. 57-73, http://hal.inria.fr/hal-00862050

[19] D. MÉRY, N. K. SINGH. *Ideal Mode Selection of a Cardiac Pacing System*, in "4th International Conference - Digital Human Modeling and applications in Health, Safety, Ergonomics and Risk Management - DHM 2013 (HCI International 2013)", Las Vegas, United States, V. G. DUFFY (editor), Lecture Notes in Computer

Science, Springer, July 2013, vol. 8025, pp. 258-267 [*DOI :* 10.1007/978-3-642-39173-6_31], http://hal.inria.fr/hal-00862077

[20] M. TOUNSI, M. MOSBAH, D. MÉRY. *From Event-B Specifications to Programs for Distributed Algorithms*, in "WETICE 2013: 22th IEEE International Conference on Enabling Technologies: Infrastructures for Collaborative Enterprises.", Hammamet, Tunisia, S. REDDY, M. JMAIEL (editors), IEEE, June 2013 [*DOI :* 10.1109/WETICE.2013.44], http://hal.inria.fr/hal-00862056

Conferences without Proceedings

[21] C. ARECES, D. DÉHARBE, P. FONTAINE, O. EZEQUIEL. *SyMT: finding symmetries in SMT formulas*, in "11th International Workshop on Satisfiability Modulo Theories - SMT", Helsinki, Finland, July 2013, http://hal.inria.fr/hal-00867816

[22] P. BAUMGARTNER, U. WALDMANN. *Hierarchic Superposition: Completeness without Compactness*, in "Fifth International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS 2013)", Nanning, China, 2013, http://hal.inria.fr/hal-00931928

[23] M. KOSTA. *SMT-Based Compiler Support for Memory Access Optimization for Data-Parallel Languages*, in "Fifth International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS 2013)", Nanning, China, 2013, http://hal.inria.fr/hal-00931958

[24] E. MABILLE, M. BOYER, L. FÉJOZ, S. MERZ. *Certifying Network Calculus in a Proof Assistant*, in "EUCASS - 5th European Conference for Aeronautics and Space Sciences", Munich, Germany, Astrium and Technische Universität München, July 2013, http://hal.inria.fr/hal-00904817

Scientific Books (or Scientific Book chapters)

[25] D. MÉRY, N. K. SINGH. *Event B*, in "Mise en oeuvre de la méthode B", J.-L. BOULANGER (editor), Informatique et Systèmes d'Informations, HERMES, April 2013, http://hal.inria.fr/hal-00926335

Books or Proceedings Editing

[26] B. CHARRON-BOST, S. MERZ, A. RYBALCHENKO, J. WIDDER (editors). , *Formal Verification of Distributed Algorithms*, Dagstuhl Reports, Dagstuhl, June 2013, vol. 3, 16 p. [*DOI :* 10.4230/DAGREP.3.4.1], http://hal.inria.fr/hal-00904805

[27] P. FONTAINE, C. RINGEISSEN, R. SCHMIDT (editors). , *Frontiers of Combining Systems*, Lecture Notes in Artificial Intelligence, Springer, September 2013, vol. 8152, 359 p. , http://hal.inria.fr/hal-00868424

References in notes

[28] J.-R. ABRIAL. , *Modeling in Event-B: System and Software Engineering*, Cambridge University Press, 2010

[29] E. ALTHAUS, E. KRUGLOV, C. WEIDENBACH. *Superposition Modulo Linear Arithmetic SUP(LA)*, in "7th Intl. Symp. Frontiers of Combining Systems (FROCOS 2009)", Trento, Italy, S. GHILARDI, R. SEBASTIANI (editors), Lecture Notes in Computer Science, Springer, 2009, vol. 5749, pp. 84-99

[30] M. ARAPINIS, M. DUFLOT. *Bounding Messages for Free in Security Protocols*, in "27th Int. Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)", Lecture Notes in Computer Science, Springer, 2007, vol. 4855, pp. 376-387

[31] L. BACHMAIR, H. GANZINGER. *Rewrite-Based Equational Theorem Proving with Selection and Simplification*, in "Journal of Logic and Computation", 1994, vol. 4, n^{o} 3, pp. 217–247

[32] R. BACK, J. VON WRIGHT. , *Refinement calculus—A systematic introduction*, Springer Verlag, 1998

[33] C. BARRETT, R. SEBASTIANI, S. A. SESHIA, C. TINELLI. *Satisfiability Modulo Theories*, in "Handbook of Satisfiability", A. BIERE, MARIJN J. H. HEULE, H. VAN MAAREN, T. WALSH (editors), Frontiers in Artificial Intelligence and Applications, IOS Press, February 2009, vol. 185, chap. 26, pp. 825-885

[34] J. C. BLANCHETTE, A. POPESCU, D. WAND, C. WEIDENBACH. *More SPASS with Isabelle - Superposition with Hard Sorts and Configurable Simplification*, in "ITP", Lecture Notes in Computer Science, Springer, 2012, vol. 7406, pp. 345-360

[35] C. W. BROWN. *Constructing a single open cell in a cylindrical algebraic decomposition*, in "Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation", New York, NY, USA, ISSAC '13, ACM, 2013, pp. 133–140

[36] T. BØGHOLM, H. KRAGH-HANSEN, P. OLSEN, B. THOMSEN, K. G. LARSEN. *Model-based schedulability analysis of safety critical hard real-time Java programs*, in "Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)", G. BOLLELLA, C. D. LOCKE (editors), ACM, 2008, pp. 106-114

[37] M. CASTRO, M. COSTA, A. ROWSTROM. *Performance and Dependability of Structured Peer-to-Peer Overlays*, in "Intl. Conf. Dependable Systems and Networks (DSN 2004)", Florence, Italy, IEEE Computer Society, 2004, pp. 9–18

[38] G. E. COLLINS. *Quantifier elimination for real closed fields by cylindrical algebraic decompostion*, in "Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20-23, 1975", H. BRAKHAGE (editor), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1975, vol. 33, pp. 134-183

[39] D. DÉHARBE, P. FONTAINE, Y. GUYOT, L. VOISIN. *SMT solvers for Rodin*, in "ABZ - Third International Conference on Abstract State Machines, Alloy, B, VDM, and Z - 2012", Pisa, Italy, J. DERRICK, J. A. FITZGERALD, S. GNESI, S. KHURSHID, M. LEUSCHEL, S. REEVES, E. RICCOBENE (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7316, pp. 194-207

[40] A. FIETZKE, E. KRUGLOV, C. WEIDENBACH. *Automatic Generation of Invariants for Circular Derivations in SUP(LA)*, in "18th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2012", N. BJØRNER, A. VORONKOV (editors), LNCS, Springer, 2012, vol. 7180, pp. 197–211

[41] A. FIETZKE, C. WEIDENBACH. *Superposition as a Decision Procedure for Timed Automata*, in "Mathematics in Computer Science", 2012, vol. 6, n^{o} 4, pp. 409-425

[42] D. JOVANOVIĆ, L. DE MOURA. *Solving Non-linear Arithmetic*, in "Automated Reasoning", B. GRAMLICH, D. MILLER, U. SATTLER (editors), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, vol. 7364, pp. 339–354

[43] E. KRUGLOV, C. WEIDENBACH. *Superposition Decides the First-Order Logic Fragment Over Ground Theories*, in "Mathematics in Computer Science", 2012, vol. 6, n^{o} 4, pp. 427-456

[44] L. LAMPORT. , *Specifying Systems*, Addison-WesleyBoston, Mass., 2002

[45] J.-Y. LE BOUDEC, P. THIRAN. , *Network Calculus*, Springer, 2001

[46] T. LU, S. MERZ, C. WEIDENBACH. *Towards Verification of the Pastry Protocol Using TLA^{+}* , in "FMOODS/FORTE", R. BRUNI, J. DINGEL (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6722, pp. 244-258

[47] S. MERZ, H. VANZETTO. *Harnessing SMT Solvers for TLA+ Proofs*, in "12th International Workshop on Automated Verification of Critical Systems (AVoCS 2012)", Bamberg, Germany, G. LÜTTGEN, S. MERZ (editors), ECEASST, EASST, December 2012, vol. 53

[48] C. MORGAN. , *Programming from Specifications*, Prentice Hall, 1998, 2nd edition

[49] D. PELED, T. WILKE. *Stutter-Invariant Temporal Properties are Expressible Without the Next-Time Operator*, in "Inf. Proc. Letters", 1997, vol. 63, n^{o} 5, pp. 243–246

[50] V. PREVOSTO, U. WALDMANN. *SPASS+T*, in "ESCoR: FLoC'06 Workshop on Empirically Successful Computerized Reasoning", Seattle, WA, USA, G. SUTCLIFFE, R. SCHMIDT, S. SCHULZ (editors), CEUR Workshop Proceedings, 2006, vol. 192, pp. 18-33

[51] V. SOFRONIE-STOKKERMANS. *Hierarchical and modular reasoning in complex theories: The case of local theory extensions*, in "Frontiers of Combining Systems. 6th International Symposium FroCos 2007, Proceedings", Liverpool, UK, B. KONEV, F. WOLTER (editors), Lecture Notes in Computer Science, Springer, 2007, vol. 4720, pp. 47-71, Invited paper