IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2014

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

1. Members **1**
2. Overall Objectives **2**
3. Research Program **2**
3.1. Automated and Interactive Theorem Proving 2
3.2. Formal Methods for Developing Algorithms and Systems 3
4. Application Domains **3**
5. New Software and Platforms **4**
5.1. The veriT Solver 4
5.2. The TLA+ Proof System 5
5.3. SPASS: An Automated Theorem Prover for First-Order Logic With Equality 5
5.4. The Redlog Computer Logic System 5
6. New Results **6**
6.1. Highlights of the Year 6
6.2. Automated and Interactive Theorem Proving 6
6.2.1. Combination of Satisfiability Procedures 6
6.2.2. Type Synthesis for Set-Theoretic Proof Obligations 7
6.2.3. Syntactic Abstractions in First-Order Modal Logics 7
6.2.4. Satisfiability of Propositional Modal Logics via SMT Solving 7
6.2.5. First-Order Extensions to Support Higher-Order Reasoning 8
6.2.6. Decidability of First-Order Recursive Clause Sets 8
6.2.7. Finite Quantification in Hierarchic Theorem Proving 8
6.2.8. Developing Learning Strategies for Virtual Substitution 8
6.2.9. Efficient Cell Construction in Cylindrical Algebraic Decomposition 9
6.3. Formal Methods for Developing Algorithms and Systems 9
6.3.1. Incremental Development of Distributed Algorithms 9
6.3.2. Modeling Medical Devices 10
6.3.3. Analysis of Real-Time Concurrent Programs 10
6.3.4. Bounding Message Length in Attacks Against Security Protocols 10
6.3.5. Evaluating and Verifying Probabilistic Systems 11
7. Bilateral Contracts and Grants with Industry **11**
7.1. Project Funded by the Airbus Foundation 11
7.2. ADN4SE Project 11
8. Partnerships and Cooperations **11**
8.1. Regional Initiatives 11
8.2. National Initiatives 12
8.2.1. ANR-DFG Project SMArT 12
8.2.2. ANR Project IMPEX 12
8.2.3. Inria Development Action VeriT 12
8.3. European Initiatives 13
8.3.1. MEALS 13
8.3.2. Cooperation with NUI Maynooth, Ireland 13
8.4. International Initiatives 13
8.5. International Research Visitors 14
9. Dissemination **14**
9.1. Promoting Scientific Activities 14
9.1.1. Organization of scientific events 14
9.1.1.1. General Chair, Scientific Chair 14
9.1.1.2. Membership in Organizing Committees 15
9.1.2. Service in Program Committees 15

9.1.2.1. Chairmanship of Conference Program Committees 15
9.1.2.2. Membership in Conference Program Committees 15
9.1.3. Journal Edition 15
9.1.4. Scientific Bodies and Self-Administration 15
9.2. Teaching - Supervision - Juries 16
9.2.1. Teaching 16
9.2.2. Supervision 16
9.2.3. Juries 16
9.3. Popularization 17
10. Bibliography **17**

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, Senior Researcher, HdR]
Uwe Waldmann [Max-Planck Institut für Informatik, Researcher]
Christoph Weidenbach [Team leader, Max-Planck Institut für Informatik, 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]

PhD Students

Manamiary Andriamiarina [Univ. de Lorraine, since Oct 2010]
Noran Azmy [Univ. des Saarlandes, since Nov 2012]
Haniel Barbosa [Inria, since Dec 2013]
Martin Bromberger [Univ. des Saarlandes, since Jul 2014]
Pablo Federico Dobal [Univ. de Lorraine, since Sep 2014 (Inria engineer until Aug 2014)]
Arnaud Fietzke [Univ. des Saarlandes, until Jun 2014]
Marek Košta [Univ. des Saarlandes, since Nov 2011]
Manuel Lamotte Schubert [Univ. des Saarlandes, since Jul 2010]
Hernán Pablo Vanzetto [Inria, until Mar 2014, Max-Planck Institut für Informatik, from May 2014]
Marco Voigt [Univ. des Saarlandes, since Nov 2013]
Daniel Wand [Univ. des Saarlandes, since Feb 2011]

Post-Doctoral Fellows

Maximilian Jaroschek [Max-Planck Institut für Informatik, granted by DFG]
Jingshu Chen [Inria, granted by Airbus Foundation and Conseil Régional de Lorraine]

Visiting Scientists

Carlos Areces [Univ. Nacional de Córdoba, Argentina, Jul 2014]
Luciana Benotti [Univ. Nacional de Córdoba, Argentina, Jul 2014]
Richard Bonichon [Univ. Federal do Rio Grande do Norte, Brazil, Sep 2014]
David Déharbe [Univ. Federal do Rio Grande do Norte, Brazil, Jan – Jul 2014]
Raúl Fervari [Univ. Nacional de Córdoba, Argentina, Jul 2014]
Guillaume Hoffmann [Univ. Nacional de Córdoba, Argentina, Jul 2014]
Claudia Tavares [Univ. Federal do Rio Grande do Norte, Brazil, Sep 2014]

Administrative Assistants

Sophie Drouot [Inria]
Delphine Hubert [Univ. de Lorraine]
Martine Kuhlmann [CNRS]
Jennifer Müller [Max-Planck Institut für Informatik]

2. Overall Objectives

2.1. Overall Objectives

The VeriDis project team includes members of the MOSEL team at 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 advances in verification techniques, including 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.

Verification techniques based on theorem proving 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 theorem proving in recent years. We are particularly interested in the integration of different deduction techniques and tools, such as automated theorem proving for relevant theories such as different fragments of arithmetic. 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 since an implementation is needed, errors are found only when they are expensive to fix. The techniques that we develop aim at automatically proving significant properties of the protocol already during 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.

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 designed for developing concurrent and distributed systems and algorithms that are firmly grounded on precise mathematical and semantical abstractions. Our common objective is to advance the state of the art in interactive and automatic deduction techniques, and their combinations, resulting in powerful tools for the computer-aided verification of distributed systems and protocols. Our techniques and tools support sound methods for the development of trustworthy distributed systems that scale to algorithms relevant for practical applications.

VeriDis members from Saarbrücken are developing SPASS [10], one of the leading automated theorem provers for first-order logic based on the superposition calculus [46]. Recent extensions to the system include the integration of dedicated reasoning procedures for specific theories, such as linear arithmetic [56], [45], 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 [57] and automated reasoning mechanisms these induce. Finally, members of the group design effective quantifier elimination methods and decision procedures for algebraic theories, supported by their efficient implementation in the Redlog system [4].

In a complementary approach to automated deduction, VeriDis members from Nancy develop veriT [1], an SMT (Satisfiability Modulo Theories [48]) 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$^+$ [52] specifications. Our prover relies on a declarative proof language, and we contribute several automatic backends [3].

3.2. Formal Methods for Developing Algorithms and Systems

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], [9], and in applying them to concrete use cases. In particular, the concept of *refinement* [44], [47], [54] in state-based modeling formalisms is central to our approach. Its basic idea is to present an algorithm or implementation through 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 establish precise proof obligations that can be discharged to a high degree 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, and they must be integrated for concrete applications.

4. Application Domains

4.1. Application Domains

Distributed algorithms and protocols are 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. Computing infrastructure must be highly available and is ideally invisible to the end user, therefore correctness is crucial. 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, the cost of unavailable services due to malfunctioning system components and software provides wider incentives for verification. For example, we are working on modeling and verifying medical devices that require closed-loop models of both the system and its environment.

5. New Software and Platforms

5.1. The veriT Solver

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

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 uninterpreted symbols, as well as a simplex-based reasoner for 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 explicit proof traces when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, non-regression tests use Inria's grid infrastructure; 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 2014 have been focused on efficiency and stability. The decision procedures for uninterpreted symbols and linear arithmetic have been further improved. There has also been some progress in the integration of the solver Redlog (section 5.4) for non-linear arithmetic in the context of the SMArT project (section 8.2).

The veriT solver participated in the SMT competition SMT-COMP 2014, part of the Vienna Summer Of Logic Olympic Games, and received the gold medal for SMT. The success of the different solvers was measured as a combination of the number of benchmark problems solved in the various categories, the number of erroneous answers, and the time taken.

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 [50]; 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: Stephan Merz [contact], Hernán Pablo Vanzetto.

TLAPS, the TLA$^+$ proof system developed at the Joint MSR-Inria Centre, is a platform for developing and mechanically verifying proofs about TLA$^+$ specifications. The TLA$^+$ proof language is hierarchical and explicit, allowing a user to decompose the overall proof into independent proof steps. TLAPS consists of a *proof manager* that interprets the proof language and generates a collection of proof obligations that are sent to *backend verifiers*. The current backends include the tableau-based prover Zenon for first-order logic, Isabelle/TLA$^+$, an encoding of TLA$^+$ as an object logic in the logical framework Isabelle, an SMT backend designed for use with any SMT-lib compatible solver, and an interface to a decision procedure for propositional temporal logic.

The current version 1.3.2 of TLAPS was released in May 2014, it is distributed under a BSD-like license at http://tla.msr-inria.inria.fr/tlaps/content/Home.html. The prover fully handles the non-temporal part of TLA$^+$. The SMT backend, developed in Nancy, has been further improved in 2014, in particular through the development of an appropriate type synthesis procedure, and is now the default backend. A new interface with a decision procedure for propositional temporal logic has been developed in 2014, so that simple temporal proof obligations can now be discharged. It is based on a technique for "coalescing" first-order subformulas of temporal logic, described in section 6.2. The standard proof library has also been further developed, partly in response to the needs of the ADN4SE project on verifying a real-time micro-kernel system (section 7.2).

TLAPS was presented at tutorials at the TLA$^+$ community event organized during ABZ 2014 in Toulouse in June and at the SPES_XT summer school at the University of Twente (The Netherlands) in September.

5.3. SPASS: An Automated Theorem Prover for First-Order Logic With Equality

Participants: Martin Bromberger, Arnaud Fietzke, Thomas Sturm, Marco Voigt, Uwe Waldmann, Christoph Weidenbach [contact].

SPASS is an automated theorem prover based on superposition that handles first-order logic with equality and several extensions for particular classes of theories. It has been developed since the mid-1990s at the Max-Planck Institut für Informatik in Saarbrücken. Version 3.7 is the current stable release; it is distributed under the FreeBSD license at http://www.spass-prover.org.

The next major release of SPASS will mainly focus on improved theory support: many applications of automated deduction require reasoning in first-order logic modulo background theories, in particular some form of arithmetic. In 2014, we have continued our efforts to improve the superposition calculus as well as to develop dedicated arithmetic decision procedures for various arithmetic theories. Our results are:

- specialized reasoning support for finite subsets,
- specialized decision procedures for linear real arithmetic with one quantifier alternation,
- new efficient and complete procedures for (mixed) linear integer arithmetic,
- decidability results and respective procedures for various combinations of linear arithmetic with first-order logic.

5.4. The Redlog Computer Logic System

Participants: Thomas Sturm [contact], Marek Košta.

Redlog is an integral part of the interactive computer algebra system Reduce. It supplements Reduce's comprehensive collection of powerful methods from symbolic computation by supplying more than 100 functions on first-order formulas. Redlog has been publicly available since 1995 and is constantly being improved. The name Redlog stands for Reduce Logic System. Andreas Dolzmann from Schloss Dagstuhl Leibniz-Zentrum is a co-developer of Redlog.

Reduce and Redlog are open-source and freely available under a modified BSD license at http://reduce-algebra.sourceforge.net/. The Redlog homepage is located at http://www.redlog.eu/. Redlog generally works with interpreted first-order logic in contrast to free first-order logic. Each first-order formula in Redlog must exclusively contain atoms from one particular Redlog-supported theory, which corresponds to a choice of admissible functions and relations with fixed semantics. Redlog-supported theories include Nonlinear Real Arithmetic (Real Closed Fields), Presburger Arithmetic, Parametric QSAT, and many more.

Effective quantifier elimination procedures for the various supported theories establish an important class of methods available in Redlog. For the theories supported by Redlog, quantifier elimination procedures immediately yield decision procedures. Besides these quantifier elimination-based decision methods there are specialized, and partly incomplete, decision methods, which are tailored to input from particular fields of application.

In 2014, Redlog made two important steps into distinct but equally important future directions. On the one hand, it integrated for the first time learning strategies, as they are known from CDCL-based SMT solving, into a classical real quantifier elimination procedure, viz. virtual substitution for linear formulas [28]. On the other hand, there was important progress concerning incomplete decision procedures for the reals. A journal submission currently under review describes identification of a Hopf bifurcation for the important MAPK model within less than a minute. The corresponding polynomial relevant for root-finding has dimension 10, total degree 100, and contains more than 850,000 monomials.

Redlog is a widely accepted tool and highly visible in mathematics, informatics, engineering and the sciences. The seminal article on Redlog [4] has received more than 300 citations in the scientific literature so far.

6. New Results

6.1. Highlights of the Year

The veriT solver (section 5.1) participated in the SMT competition 2014, part of the Vienna Summer Of Logic Olympic Games, and received the gold medal for the SMT category.

6.2. Automated and Interactive Theorem Proving

Participants: Pascal Fontaine, Marek Košta, Manuel Lamotte Schubert, Stephan Merz, Thomas Sturm, Hernán Pablo Vanzetto, Uwe Waldmann, Daniel Wand, Christoph Weidenbach.

6.2.1. Combination of Satisfiability Procedures

Joint work with Christophe Ringeissen from the CASSIS project-team at Inria Nancy Grand Est, and Paula Chocron, a student at the University of Buenos Aires.

A satisfiability problem is often expressed in a combination of theories, and a natural approach consists in solving the problem by combining the satisfiability procedures available for the component theories. This is the purpose of the combination method introduced by Nelson and Oppen. However, in its initial presentation, the Nelson-Oppen combination method requires the theories to be signature-disjoint and stably infinite (to ensure the existence of an infinite model). The design of a generic combination method for non-disjoint unions of theories is clearly a hard task but it is worth exploring simple non-disjoint combinations that appear frequently in verification. An example is the case of shared sets, where sets are represented by unary predicates. Another example is the case of bridging functions between data structures and a target theory (e.g., a fragment of arithmetic).

The notion of gentle theory has been introduced in the last few years as one solution to go beyond the restriction of stable infiniteness, in the case of disjoint theories. In [26], [43], we adapt the notion of gentle theory to the non-disjoint combination of theories sharing only unary predicates, constants, and equality. As in the disjoint case, combining two theories, one of them being gentle, requires some minor assumptions on the other one. We show that major classes of theories, i.e., Loewenheim and Bernays-Schoenfinkel-Ramsey, satisfy the appropriate notion of gentleness introduced for this particular non-disjoint combination framework.

We have also considered particular non-disjoint unions of theories connected via bridging functions [27]. We present a combination procedure which is proved correct for the theory of absolutely free data structures. We consider the problem of adapting the combination procedure to obtain a satisfiability procedure for the standard interpretations of the data structure. We present an enumeration procedure that allows us to revisit the case of lists with length.

6.2.2. Type Synthesis for Set-Theoretic Proof Obligations

TLA$^+$ is a language for the formal specification of systems and algorithms whose first-order kernel is a variant of untyped Zermelo-Fraenkel set theory. Typical proof obligations that arise during the verification of TLA$^+$ specifications mix reasoning about sets, functions, arithmetic, tuples, and records. One of the challenges in designing an efficient encoding of TLA$^+$ proof obligations for the input languages of first-order automatic theorem provers or SMT solvers is to synthesize appropriate sorts for the terms appearing in a proof obligation, matching the type system of the target prover. We base this synthesis on the detection of "typing hypotheses" present in the proof obligations and then propagate this information throughout the entire formula. An initial type system [53] similar to the multi-sorted discipline underlying SMT-lib was not expressive enough for representing constraints such as domain conditions for function applications. We therefore developed a more expressive type system that includes dependent types, predicate types, and subtyping. Type synthesis in this system is no longer decidable but generates constraints that are submitted to SMT solvers during type reconstruction. When the constraints are valid, the translation of the formula becomes simpler, and checking it becomes correspondingly more efficient. When type construction does not succeed, the translator locally falls back to a sound, but inefficient "untyped" encoding where interpreted sorts such as integers are injected into the SMT sort representing TLA$^+$ values. In practice, this approach is found to behave significantly better than the original type system, and it extends easily to ATP proof backends. The results have been published at NFM 2014 [29], full details appear in Vanzetto's PhD thesis [11].

6.2.3. Syntactic Abstractions in First-Order Modal Logics

Joint work with Damien Doligez, Jael Kriener, Leslie Lamport, and Tomer Libal within the TLA$^+$ project at the MSR-Inria Joint Centre.

TLA$^+$ proofs mix first-order and temporal logics, and few (semi-)automatic proof tools support such languages. Moreover, natural deduction and sequent calculi, which are standard underpinnings for reasoning in first-order logic, do not extend smoothly to modal or temporal logics, due to the presence of implicit parameters designating the current point of evaluation. We design a syntactic abstraction method for obtaining pure first-order, respectively propositional modal or temporal, formulas from proof obligations in first-order modal or temporal logic, and prove the soundness of this "coalescing" technique. The resulting formulas can be passed to existing automatic provers or decision procedures for first-order logic (possibly with theory support), respectively for propositional modal and temporal logic. The method is complete for proving safety properties of specifications. This work was presented at the workshop on Automated Reasoning in Quantified Non-Classical Logic organized as part of Vienna Summer of Logic [33], and it has been implemented within TLAPS (section 5.2).

6.2.4. Satisfiability of Propositional Modal Logics via SMT Solving

Joint work with Carlos Areces from the National University of Córdoba, Argentina, and Clément Herouard, a student at ENS Rennes.

Modal logics extend classical propositional logic, and they are robustly decidable. Most existing decision procedures for modal logics are based on tableau constructions. Within our ongoing cooperation with members of the National University of Córdoba supported by the MEALS and MISMT projects (sections 8.3 and 8.4), we are investigating the design of decision procedures based on adding custom instantiation rules to standard SAT and SMT solvers. Our constructions build upon the well-known standard translation of modal logics to the guarded fragment of first-order logic. The idea is to let the solver maintain an abstraction of the quantified formulas, together with corresponding models. The abstraction is refined by lazily instantiating quantifiers, until either it is found to be unsatisfiable or no new instantiations need to be considered. We prove the soundness, completeness, and termination of the procedure for basic modal logic and several extensions. In particular, a smooth extension to hybrid logic makes use of the decision procedures for equality built into SMT solvers, yielding surprisingly simple correctness proofs. A presentation of this work has been accepted for publication in 2015.

6.2.5. First-Order Extensions to Support Higher-Order Reasoning

In contrast to higher-order logic, first-order logic provides automation and completeness. In order to increase the success rate of first-order proof procedures on translations of higher-order proof obligations, we developed two extensions to first-order logic:

- a polymorphic type system and
- declarations for inductive data types.

While the former can be seen as “just some kind of complication” to standard first-order reasoning procedures, the latter is an extension beyond first-order logic. We have shown how to keep first-order completeness in the presence of inductive data types while making use of the declarations for inferences and reductions that cannot be justified at the first-order level. The result is a superposition calculus extended with induction that shows impressive performance on standard benchmark sets when compared to existing approaches.

6.2.6. Decidability of First-Order Recursive Clause Sets

Recursion is a necessary source for first-order undecidability of clause sets. If there are no cyclic, i.e., recursive definitions of predicates in such a clause set, (ordered) resolution terminates, showing decidability. In this work we present the first characterization of recursive clause sets enabling non-constant function symbols and depth increasing clauses but still preserving decidability. For this class called BDI (Bounded Depth Increase) we present a specialized superposition calculus. This work has been published in the Journal of Logic and Computation [18].

6.2.7. Finite Quantification in Hierarchic Theorem Proving

Joint work with Peter Baumgartner and Joshua Bax from NICTA, Canberra, Australia.

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. For the case when all variables occurring below such function symbols are quantified over a finite subset of their domains, we have developed and implemented a non-naive decision procedure for extended theories on top of a black-box decision procedure for the EA-fragment of the background theory. In its core, it employs a model-guided instantiation strategy for obtaining pure background formulas that are equi-satisfiable with the original formula. Unlike traditional finite model finders, it avoids exhaustive instantiation and, hence, is expected to scale better with the size of the domains [25].

6.2.8. Developing Learning Strategies for Virtual Substitution

Joint work with Konstantin Korovin from the University of Manchester, UK.

During the past twenty years there have been a number of successful applications of real quantifier elimination methods based on virtual substitution. On the other hand, recently there has been considerable progress in (linear and non-linear) real arithmetic SMT-solving triggered by the idea to adopt from Boolean SAT-solving conflict analysis and learning techniques. In this work we do the first steps towards combining these two lines of research.

We consider linear real arithmetic SMT-solving. Inspired by related work for the Fourier-Motzkin method, we develop learning strategies for linear virtual substitution. For the first time, we formalize a virtual substitution-based quantifier elimination method—with and without our learning strategies—as formal calculi in the style of abstract DPLL [55]. We prove soundness and completeness for these calculi. Some standard linear programming benchmarks computed with an experimental implementation of our calculi show that the novel learning techniques combined with linear virtual substitution give rise to considerable speedups. Our implementation is part of the Reduce package Redlog, which is open-source and freely available.

This work gave rise to a publication at the CASC 2014 international workshop [28].

6.2.9. Efficient Cell Construction in Cylindrical Algebraic Decomposition

Joint work with Christopher W. Brown from the United States Naval Academy.

In their 2012 paper, de Moura and Jovanović [51] give a novel procedure for non-linear real SMT solving. The procedure uses DPLL-style techniques to search for a satisfying assignment. In case of a conflict, Cylindrical Algebraic Decomposition (CAD) 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 that point. The crucial part of this "model-based" approach is a function realizing this cell learning. Unfortunately, it is the main computational bottleneck of the whole procedure.

In 2014, we improved our cell learning procedure developed in 2013 by further theoretical investigation, which led to optimizations of the cell construction algorithm. This work gave rise to a publication in the Journal of Symbolic Computation [14].

In this publication we present an algorithm for the cell construction problem. Given a point and a set of polynomials, the algorithm constructs a single cylindrical cell containing the point, such that the polynomials are sign-invariant in the constructed cell. To represent a single cylindrical cell, a novel data structure is introduced. The algorithm, which is based on McCallum's projection operator, works with this representation and proceeds incrementally: First a cell representing the whole real space is constructed, and then refinement with respect to a single input polynomial is done to ensure the sign-invariance of this polynomial in the refined cell. We prove that our algorithm is correct and efficient in the following sense: First, the set of polynomials computed by our algorithm is a subset of the set constructed by the "model-based" approach, and second, the cell constructed by our algorithm is bigger than the cell constructed by the "model-based" approach.

6.3. Formal Methods for Developing Algorithms and Systems

Participants: Manamiary Andriamiarina, Jingshu Chen, Marie Duflot-Kremer, Dominique Méry, Stephan Merz.

6.3.1. Incremental Development of Distributed Algorithms

Joint work with Mohammed Mosbah and Mohammed Tounsi from the LABRI laboratory in Bordeaux, France, and with Neeraj Kumar Singh from the Department of Computing and Software, McMaster University, Hamilton, Canada.

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 underlying the VISIDIA toolkit developed at LABRI for designing distributed algorithms expressed as a set of rewriting rules over graph structures.

In particular, we show how state-based models can be developed for specific problems [22] 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 [24], [12], [42]. Our patterns simplify the development of distributed systems using refinement and temporal logic.

6.3.2. Modeling Medical Devices

Formal modelling techniques and tools [30] 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 [32], we give the semantics of refinement diagrams that are used in a refinement-based methodology for complex medical systems design, which possesses all the required key features. A refinement-based approach relying on formal verification, model validation using a model-checker, and refinement charts is proposed in this methodology for designing a high-confidence medical device. We show the effectiveness of this methodology for the design of a cardiac pacemaker system. Moreover, we organized a Dagstuhl seminar on the Pacemaker Challenge [20].

6.3.3. Analysis of Real-Time Concurrent Programs

Joint work with Nadezhda Baklanova, Jan-Georg Smaus, Wilmer Ricciotti, and Martin Strecker at IRIT Toulouse, France, and master student Jorge Ibarra Delgado, funded by the Airbus Foundation (see also section 7.1).

We investigate techniques for the formal verification of multi-threaded real-time programs. We assume that programs contain annotations that indicate the times for executing basic blocks, and that these annotations are enforced by the execution platform. Inspired by Safety-Critical Java [49], our partners in Toulouse developed a formal semantics for a fragment of Java in Isabelle/HOL. We designed techniques for formally ensuring the absence of concurrent accesses to shared resources in bounded-length executions of such programs. Specifically, we generate constraints that characterize the possible execution orders of the program, and then invoke an SMT solver in order to verify that no execution violates precedence constraints that ensure absence of conflicts. In the case where such an execution exists, we obtain a trace that exhibits the access conflict. Our technique has been implemented prototypically, and appears to scale much better than a previous analysis based on an encoding of programs as timed automata. The results have been published at AVoCS 2014 [15].

During his internship within the first year of the Erasmus Mundus master program on Dependable Software Systems, Jorge Ibarra Delgado investigated the possibility of adapting the JOP toolset for Safety-Critical Java, and in particular its Worst-Case Execution Time (WCET) analyzer, for obtaining suitable annotations for basic blocks.

6.3.4. Bounding Message Length in Attacks Against Security Protocols

Joint work with Myrto Arapinis from the University of Glasgow, 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 that, under a syntactic and reasonable condition of “well-formedness” on the protocol, we can get rid of the infinitely branching part. A journal version of this result, extending the set of security properties to which it is applicable and that particular includes authentication properties, has been published in Information and Computation [13].

6.3.5. Evaluating and Verifying Probabilistic Systems

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 fulfills 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 fall outside the scope of model checking. In particular, it is often of interest 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 submitted. The first one presents the approach in details with a few illustrative applications. The second one focuses on biological applications, 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 the Airbus 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 6.3, 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.

7.2. ADN4SE Project

Participant: Stephan Merz.

Joint work with Damien Doligez of Inria Paris Rocquencourt and Jael Kriener and Tomer Libal at the Joint MSR-Inria Centre.

The ADN4SE project started in 2013 within *Programme d’Investissements d’Avenir: Briques Génériques du Logiciel Embarqué* and is coordinated for Inria by the Gallium team in Rocquencourt. The objective of this project is to develop and commercialize the PharOS real-time micro-kernel operating system. In cooperation with researchers at CEA List, we are contributing to the project by verifying key properties (in particular, determinism) of a high-level model of the system written in TLA$^+$.

8. Partnerships and Cooperations

8.1. Regional Initiatives

Participants: Jingshu Chen, Pablo Federico Dobal, Pascal Fontaine, Stephan Merz.

The PhD thesis of Pablo Federico Dobal benefits from joint funding by Région Lorraine since September 2014, complementing funding through the ANR-DFG project SMArT (section 8.2).

The post-doctoral research stay of Jingshu Chen was supported by joint funding by Région Lorraine and the Airbus Foundation.

8.2. National Initiatives

8.2.1. ANR-DFG Project SMArT

Participants: Haniel Barbosa, David Déharbe, Pablo Federico Dobal, Pascal Fontaine, Maximilian Jaroschek, Marek Košta, Stephan Merz, Thomas Sturm.

The SMArT (Satisfiability Modulo Arithmetic Theories) project is funded by *ANR-DFG Programmes blancs 2013*, a program of the Agence Nationale de la Recherche and the (German) Deutsche Forschungsgemeinschaft DFG. It started in April 2014. The partners are both the French and German parts of VeriDis and the Systerel company. The objective of the SMArT project is to provide advanced techniques for arithmetic reasoning beyond linear arithmetic for formal system verification, and particularly for SMT. Arithmetic reasoning is one strong direction of research at MPI, and the state-of-the-art tool Redlog (section 5.4) is mainly developed by Thomas Sturm. The SMT solver veriT (section 5.1), developed in Nancy, will serve as an experimentation platform for theories, techniques and methods designed within this project.

In September 2014, Pablo Federico Dobal has been hired as a PhD student in joint supervision with Saarland University, co-funded by the SMArT project and the Région Lorraine. More information on the project can be found on http://smart.gforge.inria.fr/.

8.2.2. ANR Project IMPEX

Participants: Manamiary Andriamiarina, Dominique Méry.

The ANR Project IMPEX is an INS ANR project that started in December 2013 for 4 years. It is coordinated by Dominique Méry, the other partners are IRIT/ENSEIHT, Systerel, Supelec and Telecom Sud Paris.

All software systems execute within an environment or context. Reasoning about the correct behavior of such systems is a ternary relation linking the requirements, system and context models. Formal methods are concerned with providing tool (automated) support for the synthesis and analysis of such models. These methods have quite successfully focused on binary relationships, for example: validation of a formal model against an informal one, verification of one formal model against another formal model, generation of code from a design, and generation of tests from requirements. The contexts of the systems in these cases are treated as second-class citizens: in general, the modeling is implicit and usually distributed between the requirements model and the system model. This project proposal is concerned with the explicit modeling of contexts as first-class citizens.

Several approaches aim at formalizing mathematical theories that are applicable in the formal developments of systems. These theories are helpful for building complex formalizations, expressing and reusing proof of properties. Usually, these theories are defined within contexts, that are imported and and/or instantiated. They usually represent the implicit semantics of the systems and are expressed by types, logics, algebras, etc. However, an implicit handling of contexts loses important information, and therefore is not expressive enough for ensuring that even a verified system is “correct”. As a very simple example, take two formally developed systems that are composed to exchange currency data represented by a float. This system is no longer consistent if one system refers to Euros and the other to dollars. The objective of the IMPEX project is to build explicit formal models of contextual semantics and to extend proof-based techniques for handling such a stronger semantics [23].

8.2.3. Inria Development Action VeriT

Participants: Pablo Federico Dobal, Pascal Fontaine.

Inria funded 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). Pablo Federico Dobal was 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 also contributed to the maintenance of the deltaSMT tool, which has been used by several other teams of SMT developers for debugging SMT solvers.

8.3. European Initiatives

8.3.1. 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)

Partners: Universidad de Buenos Aires, Universidad Nacional de Córdoba, Universidad Nacional de Rio Cuarto, Instituto Tecnológico Buenos Aires

Inria contact: Catuscia 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 within work package 2. In 2014, the project funded visits by Stephan Merz to Córdoba and by Carlos Areces, Luciana Benotti, Raúl Fervari, and Guillaume Hoffmann to Nancy.

8.3.2. Cooperation with NUI Maynooth, Ireland

Participant: Dominique Méry.

We cooperate with Rosemary Monahan of NUI Maynooth on exchanges between techniques of software refinement and software verification. Our cooperation was financially supported in 2013 by a one-year project funded by PHC Ulysses. 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 and provide 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 for integrating the *a posteriori* paradigm Spec# and the *a priori* paradigm Event-B. This integration induces a methodology that bridges the gap between software modeling and program verification in the software development life cycle. During 2014, we have designed the Rodin plugin EB2RC that implements transformations of Event-B models into algorithms.

8.4. International Initiatives

8.4.1. Participation In International Programs

8.4.1.1. STIC AmSud MISMT

Participants: Carlos Areces, Haniel Barbosa, Luciana Benotti, Richard Bonichon, David Déharbe, Pablo Federico Dobal, Raúl Fervari, Pascal Fontaine, Guillaume Hoffmann, Stephan Merz, Claudia Tavares.

VeriDis has a close working relationship with two South American teams at Universidade Federal do Rio Grande de Norte (UFRN), Brazil (more specifically with Prof. David Déharbe), and at Universidad Nacional de Córdoba, Argentina (more specifically with Prof. Carlos Areces). The STIC AmSud MISMT project, including both teams and VeriDis, started in 2014. It complements the MEALS project (section 8.3) and extends it to cooperation with UFRN.

The project is centered around Satisfiability Modulo Theories, with a focus on applications to Modal Logic. Notably, the project sustains the development of the veriT solver (section 5.1), of which David Déharbe and Pascal Fontaine are the main developers. First results on using SMT for modal logic have been accepted for publication.

In February, Stephan Merz spent three weeks in Córdoba. David Déharbe stayed in Nancy until July, on a sabbatical from UFRN. A workshop with many participants from the project took place in Nancy in early July. Richard Bonichon and Claudia Tavares visited Nancy in September. At the end of the year, Haniel Barbosa (VeriDis PhD student in joint supervision with Natal) spent three months in Natal and visited Córdoba for two weeks.

More information on the STIC AmSud MISMT project is available on http://mismt.gforge.inria.fr/.

8.5. International Research Visitors

8.5.1. Visits of International Scientists

David Déharbe from UFRN (Natal, Brazil) spent a sabbatical year with the VeriDis team in Nancy from August, 2013 to July, 2014.

8.5.1.1. Internships

Ignacio Martin Queralt

Subject: Symbolic transition checking for TLA$^+$

Date: April to September, 2014

Institution: Universidad Nacional de Córdoba (Argentina)

Clément Herouard

Subject: SMT techniques for modal logics and extensions

Date: May to July, 2014

Institution: Ecole Normale Supérieure de Rennes (France)

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Organization of scientific events

9.1.1.1. General Chair, Scientific Chair

Pascal Fontaine co-organized the SAT/SMT Summer School 2014, affiliated with Vienna Summer of Logic, in Semmering, Austria.

Stephan Merz is a member of the steering committee of the workshop on Automated Verification of Critical Systems (AVoCS). He co-organized the TLA$^+$ Community Event, a satellite of ABZ 2014 in June in Toulouse.

Thomas Sturm is chair of the steering committee of the conference series *Mathematical Aspects of Computer and Information Sciences* (MACIS). He coordinated the CDZ Workshop GZ1115 *Computation and Reasoning with Constraints*, Beijing, China, 2014.

Christoph Weidenbach is a member of the steering committee of *Bundeswettbewerb Informatik*, the German competition among high-school students in computer science.

9.1.1.2. Membership in Organizing Committees

The International Summer School on Verification Techniques, Systems, and Applications (VTSA) has been organized since 2008 in the Greater Region (Nancy, Saarbrücken, Luxembourg, and Liège), and Stephan Merz and Christoph Weidenbach are co-organizers of VTSA. In 2014, VTSA took place during the last week of October in Luxembourg.

9.1.2. Service in Program Committees

9.1.2.1. Chairmanship of Conference Program Committees

Dominique Méry was co-chair of the program committee of the 11th International Colloquium on Theoretical Aspects of Computing, held in Bucharest, Romania in September, and of the First Workshop on Formal Integrated Development Environments, a satellite of ETAPS in Grenoble, France, in April.

Stephan Merz was co-chair of the program committee of the 16th International Conference on Formal Engineering Methods, held in Luxembourg in November, and of the First International Workshop on Formal Reasoning in Distributed Algorithms (FRIDA) in July, as part of Vienna Summer of Logic.

Christoph Weidenbach was co-chair of the program committee for the 7th International Joint Conference on Automated Reasoning (IJCAR) that took place in Vienna, Austria, as part of Vienna Summer of Logic.

9.1.2.2. Membership in Conference Program Committees

Pascal Fontaine served on the program committees of the International Joint Conference on Automated Reasoning (IJCAR) and of the workshops PAAR and SMT.

Dominique Méry served on the program committees of ABZ, AFADL, CSDM, MedicalCPS, FHIES, FM, ICFEM, iFM, and FACS.

Stephan Merz served on the program committees of the international conferences ABZ, IJCAR, SAC, SEFM, SSS, and of the workshops AVoCS, GRSRD, SETS, and VERIFY.

Uwe Waldmann served on the program committee of IJCAR.

9.1.3. Journal Edition

Stephan Merz (together with Gerald Lüttgen of University of Bamberg) edited a special issue of Science of Computer Programming on the Automated Verification of Critical Systems, following AVoCS 2012.

Thomas Sturm is a member of the editorial boards of the *Journal of Symbolic Computation* (Elsevier) and *Mathematics in Computer Science* (Springer).

Christoph Weidenbach is an editor of the Journal of Automated Reasoning.

9.1.4. Scientific Bodies and Self-Administration

Pascal Fontaine was an elected member of the SMT Steering Committee (2012–2014), and he is one of three SMT-LIB managers. He was elected CADE trustee in October 2014.

Dominique Méry is a member of the IFIP Working Group 1.3 on *Foundations of System Specification*. He is the head of the Doctoral School IAEM Lorraine for the University of Lorraine and head of the Formal Methods department of the LORIA laboratory. He is an expert for the French Ministry of Education (DS9), an expert for the French Agence Nationale de la Recherche (ANR) and for AERES.

Stephan Merz is a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*. He is a member of the Scientific Directorate of the International Computer Science Meeting Center in Schloss Dagstuhl. He is the delegate for the organization of conferences at the Inria Nancy Grand Est research center and co-head of the PhD committee for computer science in Lorraine. He was a member of the hiring committee for an associated professor at Université Toulouse 3 and an expert for Agence Nationale de la Recherche (ANR), for Haut Conseil de l'Évaluation de la Recherche et de l'Enseignement Supérieur (HCERES), for the German DFG, the Dutch NWO, and for the European Research Council (ERC).

Thomas Sturm is a member of the selection committee for MSc and PhD students at the International Max Planck Research School.

Christoph Weidenbach is a trustee of CADE (elected 2009, reelected 2012). He is a member of the selection committee of the Saarbrücken Graduate School in Computer Science.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

The university employees of VeriDis (in Nancy) have statutory teaching obligations of 192 hours per year. We indicate the graduate courses that members of the team have been teaching in 2014.

Marie Duflot-Kremer taught a course on Introduction to Algorithmic Verification (first-year master level at Université de Lorraine). She and Stephan Merz also taught a course on Algorithmic Verification in the second year of master and for students of Erasmus Mundus Dependable Software Systems.

Pascal Fontaine is head of the Master MIAGE (Business Informatics) at Université de Lorraine since September 2014.

Stephan Merz taught a course on formal specification using TLA$^+$ at the SPES_XT summer school on model-based development of embedded systems at the University of Twente (The Netherlands) in September.

Uwe Waldmann gave courses on Automated Reasoning I and II at Saarland University.

Christoph Weidenbach gave a course on Automated Reasoning I.

9.2.2. Supervision

PhD: Arnaud Fietzke, Labelled Superposition, Universität des Saarlandes. Supervised by Christoph Weidenbach, defended on June 5, 2014.

PhD: Hernán Pablo Vanzetto, Proof Automation and Type Synthesis for Set Theory in the Context of TLA$^+$, Université de Lorraine. Supervised by Kaustuv Chaudhuri and Stephan Merz, defended on December 8, 2014.

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 Stephan Merz and Christoph Weidenbach, since 11/2012.

PhD in progress: Haniel Barbosa, Refutational Completeness in Satisfiability Modulo Theories, Université de Lorraine and UFRN (Natal, Brazil). Supervised by David Déharbe, Pascal Fontaine, and Stephan Merz, since 12/2013.

PhD in progress: Martin Bromberger, Arithmetic Reasoning, Saarland University. Supervised by Christoph Weidenbach, since 07/2014.

PhD in progress: Pablo Federico Dobal, Satisfiability Modulo Arithmetic Theories, Université de Lorraine and Saarland University. Supervised by Pascal Fontaine, Stephan Merz, and Thomas Sturm, since 09/2014.

PhD in progress: Marek Košta, Computational Logic, Universität des Saarlandes. Supervised by Thomas Sturm, since 11/2011.

PhD in progress: Manuel Lamotte Schubert, Automatic Authorization Analysis, Saarland University. Supervised by Christoph Weidenbach, 07/2010.

PhD in progress: Marco Voigt, Decidable Hierarchic Combinations, Saarland University. Supervised by Christoph Weidenbach, since 11/2013.

PhD in progress: Daniel Wand, First-Order Extensions to Support Higher-Order Reasoning, Saarland University. Supervised by Christoph Weidenbach, since 02/2011.

9.2.3. Juries

Stephan Merz served as a reviewer for the PhD theses of Nadezhda Baklanova (Univ. Toulouse 3), Claire Dross (Univ. Paris Sud), and Giuliano Losa (EPFL Lausanne).

9.3. Popularization

Marie Duflot-Kremer took part in various popularization activities, with a public ranging from primary school kids (with unplugged activities concerning sorting, programming, error detection) to non-scientific staff of the Inria center. She is also a member of the steering committee preparing an itinerant exposition intended for explaining computer science to high-school students and took part in an event of the European Code Week in Paris.

Pascal Fontaine and Stephan Merz illustrated some subjects and techniques that underly formal verification of protocols and algorithms at events like “Fête de la Science”. Using wooden puzzles and Sudoku sheets, they explained how real-life problems can be represented in logical form and then solved using automated tools based on formal logic.

Christoph Weidenbach lectured within the series “Perspektiven der Informatik” at Saarland University and within the public lecture series of the federal state of Saarland.

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] A. Dolzmann, T. Sturm. *Redlog: Computer algebra meets computer logic*, in "ACM SIGSAM Bull.", 1997, vol. 31, n^{o} 2, pp. 2-9

[5] 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

[6] 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

[7] 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

[8] 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/

[9] 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

[10] 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

Doctoral Dissertations and Habilitation Theses

[11] H. VANZETTO. *Proof automation and type synthesis for set theory in the context of TLA+*, Université de Lorraine, December 2014, https://hal.inria.fr/tel-01096518

Articles in International Peer-Reviewed Journals

[12] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Revisiting Snapshot Algorithms by Refinement-based Techniques (Extended Version)*, in "Computer Science and Information Systems", January 2014, vol. 11, n^{o} 1, pp. 251-270 [*DOI :* 10.2298/CSIS130122007A], https://hal.inria.fr/hal-00924525

[13] M. ARAPINIS, M. DUFLOT. *Bounding messages for free in security protocols – extension to various security properties*, in "Information and Computation", 2014, 34 p. [*DOI :* 10.1016/J.IC.2014.09.003], https://hal.inria.fr/hal-01083657

[14] C. W. BROWN, M. KOSTA. *Constructing a single cell in cylindrical algebraic decomposition*, in "Journal of Symbolic Computation", September 2014, 35 p. , https://hal.inria.fr/hal-01088452

[15] J. CHEN, M. DUFLOT, S. MERZ. *Analyzing Conflict Freedom For Multi-threaded Programs With Time Annotations*, in "Electronic Communications of the EASST", December 2014, vol. 70, 14 p. , https://hal.inria.fr/hal-01087871

[16] D. DÉHARBE, P. FONTAINE, L. VOISIN, Y. GUYOT. *Integrating SMT solvers in Rodin*, in "Science of Computer Programming", November 2014, vol. 94, 14 p. , https://hal.inria.fr/hal-01094999

[17] M. KOSTA, P. DURIS. *Flip-Pushdown Automata with k Pushdown Reversals and E0L Systems are Incomparable*, in "Information Processing Letters", 2014, vol. 114, n^{o} 8, pp. 417-420, https://hal.inria.fr/hal-01088446

[18] M. LAMOTTE-SCHUBERT, C. WEIDENBACH. *BDI: a new decidable clause class*, in "Journal of Logic and Computation", 2014, vol. 24, n^{o} 6, 28 p. , https://hal.inria.fr/hal-01098084

[19] G. LÜTTGEN, S. MERZ. *Editorial: Special Issue of Automated Verification of Critical Systems*, in "Science of Computer Programming", December 2014, vol. 96, n^{o} 3, pp. 277-278, https://hal.inria.fr/hal-01084232

[20] D. MÉRY, B. SCHÄTZ, A. WASSYNG. *The Pacemaker Challenge: Developing Certifiable Medical Devices (Dagstuhl Seminar 14062)*, in "Dagstuhl Reports", 2014, vol. 4, n^{o} 2, pp. 17–37, https://hal.inria.fr/hal-01097629

Invited Conferences

[21] C. BARRETT, L. DE MOURA, P. FONTAINE. *Proofs in satisfiability modulo theories*, in "APPA (All about Proofs, Proofs for All)", Vienna, Austria, July 2014, https://hal.inria.fr/hal-01095009

[22] D. MÉRY. *Playing with State-Based Models for Designing Better Algorithms*, in "MEDI - Model and Data Engineering - 4th International Conference", Larrnaca, Greece, Y. A. AMEUR, L. BELLATRECHE, G. A. PAPADOPOULOS (editors), Lecture Notes in Computer Science, Springer, September 2014, vol. 8748, pp. 1-3, https://hal.inria.fr/hal-01097625

International Conferences with Proceedings

[23] Y. A. AMEUR, J. P. GIBSON, D. MÉRY. *On Implicit and Explicit Semantics: Integration Issues in Proof-Based Development of Systems*, in "Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications - 6th International Symposium,", Corfu, Greece, T. MARGARIA, B. STEFFEN (editors), Lectures Notes in Computer Science, Springer, October 2014, vol. 8803, pp. 604-618, https://hal.inria.fr/hal-01097624

[24] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Analysis of Self-* and P2P Systems using Refinement*, in "ABZ 2014 - 4th International ABZ 2014 Conference ASM, Alloy, B, TLA, VDM, Z", Toulouse, France, Y. AIT AMEUR, K.-D. SCHEWE (editors), LNCS, Springer, June 2014, vol. 8477, pp. 117-123 [*DOI :* 10.1007/978-3-662-43652-3_9], https://hal.inria.fr/hal-01018125

[25] P. BAUMGARTNER, J. BAX, U. WALDMANN. *Finite Quantification in Hierarchic Theorem Proving*, in "7th International Joint Conference on Automated Reasoning (IJCAR 2014)", Vienna, Austria, S. DEMRI, D. KAPUR, C. WEIDENBACH (editors), Lecture Notes in Computer Science, Springer, July 2014, vol. 8562, pp. 152-167, https://hal.inria.fr/hal-01087873

[26] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *A Gentle Non-Disjoint Combination of Satisfiability Procedures*, in "Automated Reasoning - 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic", Vienna, Austria, Lecture Notes in Computer Science, Springer, July 2014, vol. 8562, pp. 122-136 [*DOI :* 10.1007/978-3-319-08587-6_9], https://hal.inria.fr/hal-01087162

[27] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *Satisfiability Modulo Non-Disjoint Combinations of Theories Connected via Bridging Functions*, in "Workshop on Automated Deduction: Decidability, Complexity, Tractability, ADDCT 2014. Held as Part of the Vienna Summer of Logic, affiliated with IJCAR 2014 and RTA 2014", Vienna, Austria, Silvio Ghilardi, Ulrike Sattler, Viorica Sofronie-Stokkermans, July 2014, https://hal.inria.fr/hal-01087218

[28] K. KOROVIN, M. KOSTA, T. STURM. *Towards Conflict-Driven Learning for Virtual Substitution*, in "CASC - Computer Algebra in Scientific Computing - 16th International Workshop", Warsaw, Poland, V. P. GERDT, W. KOEPF, W. M. SEILER, E. V. VOROZHTSOV (editors), Lecture Notes in Computer Science, Springer, 2014, vol. 8660, pp. 256-270, https://hal.inria.fr/hal-01088450

[29] S. MERZ, H. VANZETTO. *Refinement Types for TLA+*, in "NASA Formal Methods - 6th International Symposium", Houston, Texas, United States, J. M. BADGER, K. Y. ROZIER (editors), LNCS, Springer, 2014, vol. 8430, pp. 143-157 [*DOI :* 10.1007/978-3-319-06200-6_11], https://hal.inria.fr/hal-01063516

[30] D. MÉRY, N. K. SINGH. *Formal Evaluation of Landing Gear System*, in "SoICT - Fifth Symposium on Information and Communication Technology", HANOI, Vietnam, N. H. SON, Y. DEVILLE, M. BUI (editors), ACM, December 2014, https://hal.inria.fr/hal-01097645

[31] D. MÉRY, N. K. SINGH. *Modeling an Aircraft Landing System in Event-B*, in "ABZ 2014 Case Study Track", Toulouse, France, F. BONIOL (editor), CCIS, Springer, June 2014, vol. 433, pp. 154-159, https://hal.inria.fr/hal-00985010

[32] D. MÉRY, N. K. SINGH. *The Semantics of Refinement Chart*, in "HCI International", Heraklion, Greece, V. G. DUFFY (editor), Lecture Notes in Computer Science, Springer, June 2014, vol. 8529, pp. 415-426 [*DOI :* 10.1007/978-3-319-07725-3_42], https://hal.inria.fr/hal-00995176

Conferences without Proceedings

[33] D. DOLIGEZ, J. KRIENER, L. LAMPORT, T. LIBAL, S. MERZ. *Coalescing: Syntactic Abstraction for Reasoning in First-Order Modal Logics*, in "ARQNL 2014 - Automated Reasoning in Quantified Non-Classical Logics", Vienna, Austria, July 2014, https://hal.inria.fr/hal-01063512

[34] K. KOROVIN, M. KOSTA, T. STURM. *Towards Conflict-Driven Learning for Virtual Substitution*, in "SMT - 12th International Workshop on Satisfiability Modulo Theories", Vienna, Austria, Informal CEUR Workshop Proceedings, Philipp Rümmer and Christoph M. Wintersteiger, July 2014, https://hal.inria.fr/hal-01088458

[35] M. KOSTA, T. STURM, A. DOLZMANN. *Better Answers to Real Questions*, in "SMT - 12th International Workshop on Satisfiability Modulo Theories", Vienna, Austria, Informal CEUR Workshop Proceedings, Philipp Rümmer and Christoph M. Wintersteiger, July 2014, 69 p. , https://hal.inria.fr/hal-01088456

[36] D. WAND. *Polymorphic+Typeclass Superposition*, in "4th Workshop on Practical Aspects of Automated Reasoning (PAAR 2014)", Vienna, Austria, B. KONEV, L. DE MOURA, S. SCHULZ (editors), July 2014, 15 p. , https://hal.inria.fr/hal-01098078

Scientific Books (or Scientific Book chapters)

[37] S. DEMRI, D. KAPUR, C. WEIDENBACH. *Automated Reasoning – Seventh International Joint Conference (IJCAR 2014)*, Lecture Notes in Computer Science, Springer, 2014, vol. 8562, https://hal.inria.fr/hal-01098072

[38] S. MERZ. *Science of Computer Programming Special Issue: Automated Verification of Critical Systems*, Science of Computer Programming, Elsevier, December 2014, vol. 96, n^{o} 3, https://hal.inria.fr/hal-01084228

[39] S. MERZ, J. PANG. *Formal Methods and Software Engineering – 16th International Conference on Formal Engineering Methods (ICFEM 2014)*, Lecture Notes in Computer Science, Springer, November 2014, vol. 8829, 460 p. , https://hal.inria.fr/hal-01098238

Books or Proceedings Editing

[40] G. CIOBANU, D. MÉRY (editors). *Theoretical Aspects of Computing – ICTAC 2014*, Lecture Notes in Computer Science, SpringerBucharest, Romania, September 2014, vol. 8687, https://hal.inria.fr/hal-01097627

[41] C. DUBOIS, D. GIANNAKOPOULOU, D. MÉRY (editors). *Proceedings 1st Workshop on Formal Integrated Development Environment*, Electronic Proceedings in Theoretical Computer Science, EPTCSFrance, April 2014, vol. 149, 105 p. [*DOI :* 10.4204/EPTCS.149.9], https://hal.inria.fr/hal-00987531

Research Reports

[42] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Analysis of Self-* and P2P Systems using Refinement (Full Report)*, 2014, https://hal.inria.fr/hal-01018162

[43] P. Chocron, P. Fontaine, C. Ringeissen. *A Gentle Non-Disjoint Combination of Satisfiability Procedures (Extended Version)*, April 2014, n$^{\text{o}}$ RR-8529, https://hal.inria.fr/hal-00985135

References in notes

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

[45] 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

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

[47] R. Back, J. von Wright. *Refinement calculus—A systematic introduction*, Springer Verlag, 1998

[48] 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

[49] 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

[50] 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

[51] 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

[52] L. Lamport. *Specifying Systems*, Addison-WesleyBoston, Mass., 2002

[53] 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

[54] C. Morgan. *Programming from Specifications*, Prentice Hall, 1998, 2nd edition

[55] R. Nieuwenhuis, A. Oliveras, C. Tinelli. *Solving SAT and SAT Modulo Theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T)*, in "J. ACM", 2006, vol. 53, n$^{\text{o}}$ 6, pp. 937-977

[56] 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

[57] 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