IN PARTNERSHIP WITH:
CNRS

Max Planck Institut für Informatik de Saarbrücken

Université de Lorraine

Activity Report 2016

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

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

Keywords:

Computer Science and Digital Science:

2.1.7. - Distributed programming
2.1.11. - Proof languages
2.4. - Verification, reliability, certification
2.4.1. - Analysis
2.4.2. - Model-checking
2.4.3. - Proofs
7.4. - Logic in Computer Science
7.6. - Computer Algebra

Other Research Topics and Application Domains:

6.1. - Software industry
6.3.2. - Network protocols
6.6. - Embedded systems

1. Members

Research Scientists

Jasmin Christian Blanchette [Inria, Starting Research Position]
Stephan Merz [Team leader, Inria, Senior Researcher, HDR]
Thomas Sturm [CNRS, Senior Researcher]
Uwe Waldmann [Max-Planck Institut für Informatik, Senior Researcher]
Christoph Weidenbach [Team leader, Max-Planck Institut für Informatik, Senior Researcher, HDR]

Faculty Members

Marie Duflot-Kremer [Univ. Lorraine, Associate Professor]
Pascal Fontaine [Univ. Lorraine, Associate Professor]
Dominique Méry [Univ. Lorraine, Professor, HDR]
Martin Strecker [Univ. Toulouse III, Associate Professor, Inria secondment since Sep 2016]

Engineers

Gabriel Corona [Univ. Lorraine, until Aug 2016]
Simon Cruanes [Inria]
Matthieu Nicolas [Inria]
Martin Riener [Inria, MSR-Inria Joint Centre]

PhD Students

Gabor Alági [Univ. des Saarlandes, since Nov 2012]
Noran Azmy [Univ. des Saarlandes, until Aug 2016]
Haniel Barbosa [Univ. Lorraine, since Dec 2013]
Martin Bromberger [Univ. des Saarlandes, since Jul 2014]
Margaux Duroeulx [Univ. Lorraine, since Oct 2016]
Mathias Fleury [Univ. des Saarlandes, since Oct 2015]
Souad Kherroubi [Univ. Lorraine, since Jan 2015]
Marek Košta [Univ. des Saarlandes, until Oct 2016]

Nicolas Schnepf [Inria, since Oct 2016]
Marco Voigt [Univ. des Saarlandes, since Nov 2013]
Daniel Wand [Univ. des Saarlandes, since Feb 2011]

Administrative Assistants

Sophie Drouot [Inria]
Martine Kuhlmann [CNRS]
Jennifer Müller [Max-Planck Institut für Informatik]

Others

Ilina Stoilkovska [TU Wien, visitor, Sep–Oct 2016]
Tung Vu Xuan [JAIST, visitor, since May 2016]

2. Overall Objectives

2.1. Overall Objectives

The VeriDis project team includes members of the MOSEL group at LORIA, the computer science laboratory in Nancy, and members of the research group *Automation of Logic* at Max-Planck-Institut für Informatik in Saarbrücken. It is headed by Stephan Merz and Christoph Weidenbach. VeriDis was created in 2010 as a local research group 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 and analysis of concurrent and distributed algorithms and systems, within the framework of mathematically precise and practically applicable development methods. We intend to assist designers of algorithms and systems in 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. We have, however, observed significant 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 than what is available in today's verification tools.

VeriDis aims at exploiting and further developing automation in system verification, and at applying 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 for 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 at moving current research in this area 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 an executable, whose production 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 mainly target designs and algorithms at high levels of abstraction; we aim at 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 gathers experts in techniques and tools for automatic deduction and interactive theorem proving, and specialists in methods and formalisms designed for the development of trustworthy concurrent and distributed systems and algorithms. Our common objective is twofold: first, we wish to advance the state of the art in automated and interactive theorem proving, and their combinations. Second, we work on making the resulting technology available for the computer-aided verification of distributed systems and protocols. In particular, our techniques and tools are intended to 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 [39]. The group also studies general frameworks for the combination of theories such as the locality principle [52] and automated reasoning mechanisms these induce.

In a complementary approach to automated deduction, VeriDis members from Nancy work on techniques for integrating reasoners for specific theories. They develop veriT [1], an SMT (Satisfiability Modulo Theories [41]) 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.

Finally, VeriDis members design effective quantifier elimination methods and decision procedures for algebraic theories, supported by their efficient implementation in the Redlog system [4].

An important objective of this line of work is the integration of theories in automated deduction. Typical theories of interest, including fragments of arithmetic, are not expressible in first-order logic. We therefore explore efficient, modular techniques for integrating semantic and syntactic reasoning methods, develop novel combination results and techniques for quantifier instantiation. These problems are addressed from both sides, e.g. by embedding a decision procedure into the superposition framework or by allowing an SMT solver to accept axiomatizations for plug-in theories. We also develop specific decision procedures for theories such as non-linear real arithmetic that are important when reasoning about certain classes of (e.g., real-time) systems but that also have interesting applications beyond verification.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction when fully automatic verification is not (yet) feasible. An interactive proof platform should help verification engineers lay out the proof structure at a sufficiently high level of abstraction; powerful automatic plug-ins should then discharge the resulting proof steps. 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 Microsoft Research-Inria Centre in Saclay on the development of methods and tools for the formal proof of TLA$^+$ [45] specifications. Our prover relies on a declarative proof language, and calls upon several automatic backends [3]. Trust in the correctness of the overall proof can be ensured when the backends provide justifications that can be checked by the trusted kernel of a proof assistant. During the development of a proof, most obligations that are passed to the prover actually fail – for example, because necessary information is not present in the context or because the invariant is too weak, and we are interested in explaining failed proof attempts to the user, in particular through the construction of counter-models.

3.2. Formal Methods for Developing and Analyzing Algorithms and Systems

Theorem provers are not used in isolation, but they support the application of 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* [38], [40], [48] in state-based modeling formalisms is central to our approach because it allows us to present a rational (re)construction of system development. An important goal in designing such methods is to establish precise proof obligations many of which can be discharged 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.

In this line of work, we investigate specific development and verification patterns for particular classes of algorithms, in order to reduce the work associated with their verification. We are also interested in applications of formal methods and their associated tools to the development of systems that underlie specific certification requirements in the sense of, e.g., Common Criteria. Finally, we are interested in the adaptation of model checking techniques for verifying actual distributed programs, rather than high-level models.

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 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 the verification of algorithms that are developed for supporting novel computing paradigms, including 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 we work together with domain experts on designing formal models of these protocols, and on verifying their properties. These cooperations help us focus on concrete algorithms and ensure that our work is relevant to the distributed algorithm community.

Formal verification techniques can contribute to certifying 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 encourage the use of formal methods. 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. Highlights of the Year

5.1. Highlights of the Year

Jasmin Blanchette was awarded an ERC Starting Grant for his Matryoshka project aiming at fast interactive verification through strong automation for higher-order constructs.

As part of a European network, Pascal Fontaine and Thomas Sturm participate in a new H2020 Coordination and Support Action. [1] In accordance with the distributed character of Veridis, we are operating nodes at LORIA as well as MPI. Further nodes are located in Austria (University of Linz), Germany (RWTH Aachen; University of Kassel), Italy (Fondazione Bruno Kessler; University of Genova), and the UK (Universties of Bath, Coventry, and Oxford; Maplesoft Europe Ltd.). The CSA aims at improving the integration of communities, methods, and software from SMT solving and symbolic computation [20].

Jasmin Blanchette and Stephan Merz were PC chairs and organizers of the 7th International Conference on Interactive Theorem Proving in Nancy (August 22–27), the main conference of developers and users of proof assistants.

5.1.1. Awards

Mathias Fleury, together with his two supervisors, received the Best Paper Award at IJCAR 2016 for their work on a formalized SAT solver.

Together with Andrew J. Reynolds at the University of Iowa, Jasmin Blanchette was invited to submit a short version of his CADE 2015 paper on a decision procedure for (co)datatypes to the Sister Conference Best Paper Track of IJCAI 2016.

BEST PAPERS AWARDS:

[25]
J. C. BLANCHETTE, M. FLEURY, C. WEIDENBACH. *A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality*, in "8th International Joint Conference on Automated Reasoning (IJCAR 2016)", Coimbra, Portugal, Automated Reasoning - 8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, June 27 - July 2, 2016, Proceedings, June 2016 [*DOI :* 10.1007/978-3-319-40229-1_4], https://hal.inria.fr/hal-01336074

[19]
A. REYNOLDS, J. C. BLANCHETTE. *A Decision Procedure for (Co)datatypes in SMT Solvers*, in "IJCAI 2016", New York City, United States, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, July 2016, https://hal.inria.fr/hal-01397082

6. New Software and Platforms

6.1. The Nunchaku Higher-Order Model Finder

FUNCTIONAL DESCRIPTION

Nunchaku is a model finder for higher-order logic, with dedicated support for various definitional principles. It is designed to work as a backend for various proof assistants and to use state-of-the-art model finders and other solvers as backends.

In 2016, the first three versions of the tools were released (0.1 through 0.3). The Isabelle2016-1 release includes Nunchaku as well as the frontend that bridges the gap between the proof assistant and the model finder. Work has commenced on a Coq frontend [28] and a TLA$^+$ frontend. Currently, the backends CVC4, Kodkod, and Paradox are supported.

- Participants: Jasmin Blanchette and Simon Cruanes
- Contact: Jasmin Blanchette
- URL: https://github.com/nunchaku-inria

[1] H2020-FETOPEN-2015-CSA-712689, http://www.sc-square.org/

6.2. The Redlog Computer Logic System

FUNCTIONAL DESCRIPTION

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 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.

In 2016 there was significant progress with the generation of models for real satisfiability problems [15]. When obtained via quantifier elimination by virtual substitutions, such models contain in general non-standard numbers like positive infinitesimal and infinite values. In an efficient post-processing step Redlog now generates standard models.

- Participants: Thomas Sturm and Marek Kosta
- Contact: Thomas Sturm
- URL: http://www.redlog.eu/

6.3. The SPASS automated theorem prover

FUNCTIONAL DESCRIPTION

The classic SPASS is an automated theorem prover based on superposition that handles first-order logic with equality and several extensions for particular classes of theories. With version SPASS 3.9 we have stopped the development of the classic prover and have started the bottom-up development of SPASS 4.0 that will actually be a workbench of automated reasoning tools.

In 2016 we have made available for the first time our LIA solver SPASS-IQ. Furthermore, we have developed a state-of-the-art SAT solver SPASS-SATT that will be available in 2017.

- Contact: Christoph Weidenbach
- URL: http://www.spass-prover.org/

6.4. TLAPS, the TLA+ Proof System

FUNCTIONAL DESCRIPTION

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 1.4.3 of TLAPS was released in June 2015, it is distributed under a BSD-like license. The prover fully handles the non-temporal part of TLA$^+$. Basic temporal logic reasoning is supported through an interface with a decision procedure for propositional temporal logic that performs on-the-fly abstraction of first-order subformulas. Symmetrically, subformulas whose main operator is a connective of temporal logic are abstracted before being sent to backends for first-order logic.

A complete rewrite of the proof manager is ongoing. Its objectives are a cleaner interaction with the standard TLA$^+$ front-ends, in particular SANY, the standard parser and semantic analyzer. This is necessary for extending the scope of the fragment of TLA$^+$ that is handled by TLAPS, such as full temporal logic and module instantiation.

TLAPS has been used in several case studies, including the proof of determinacy of PharOS [21] and the verification of the Pastry routing protocol [22]. These case studies feed back into the development of the proof system and of its standard library.

- Contact: Stephan Merz
- URL: https://tla.msr-inria.inria.fr/tlaps/content/Home.html

6.5. The veriT Solver

Scientific Description

veriT comprises a SAT solver, a congruence closure based decision procedure for uninterpreted symbols, a simplex-based decision procedure for linear arithmetic, and instantiation-based quantifier handling.

Functional Description

VeriT is an open, trustable and efficient SMT (Satisfiability Modulo Theories) solver, featuring efficient decision procedure for uninterpreted symbols and linear arithmetic, and quantifier reasoning.

Efforts in 2016 have been focused on non-linear arithmetic reasoning and quantifier handling. The reasoning capabilities of veriT have been significantly improved along those two axes, but non-linear arithmetic reasoning is not yet stable.

The veriT solver participated in the SMT competition SMT-COMP 2016 with good results.

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, it is integrated within the Atelier B.

- Participants: Pascal Fontaine, David Déharbe and Haniel Barbosa
- Partners: Université de Lorraine - Federal University of Rio Grande do Norte
- Contact: Pascal Fontaine
- URL: http://www.veriT-solver.org

7. New Results

7.1. Automated and Interactive Theorem Proving

Participants: Gabor Alági, Haniel Barbosa, Jasmin Christian Blanchette, Martin Bromberger, Simon Cruanes, Mathias Fleury, Pascal Fontaine, Marek Košta, Stephan Merz, Martin Riener, Martin Strecker, Thomas Sturm, Marco Voigt, Uwe Waldmann, Daniel Wand, Christoph Weidenbach.

7.1.1. IsaFoL: Isabelle Formalization of Logic

Joint work with Heiko Becker (MPI-SWS Saarbrücken), Peter Lammich (TU München), Andrei Popescu (Middlesex University London), Anders Schlichtkrull (DTU Copenhagen), Dmitriy Traytel (ETH Zürich), and Jørgen Villadsen (DTU Copenhagen).

Researchers in automated reasoning spend a significant portion of their work time specifying logical calculi and proving metatheorems about them. These proofs are typically carried out with pen and paper, which is error-prone and can be tedious. As proof assistants are becoming easier to use, it makes sense to employ them.

In this spirit, we started an effort, called IsaFoL (Isabelle Formalization of Logic), that aims at developing libraries and methodology for formalizing modern research in the field, using the Isabelle/HOL proof assistant.[2] Our initial emphasis is on established results about propositional and first-order logic. In particular, we are formalizing large parts of Weidenbach's forthcoming textbook, tentatively called *Automated Reasoning—The Art of Generic Problem Solving*.

The objective of formalization work is not to eliminate paper proofs, but to complement them with rich formal companions. Formalizations help catch mistakes, whether superficial or deep, in specifications and theorems; they make it easy to experiment with changes or variants of concepts; and they help clarify concepts left vague on paper.

The repository contains six completed entries and three entries that are still in development. Notably:

- Mathias Fleury formalized a SAT solver framework with learn, forget, restart, and incrementality and published the result at a leading conference, together with Jasmin Blanchette and Christoph Weidenbach [25].
- Anders Schlichtkrull, remotely co-supervised by Jasmin Blanchette, formalized unordered first-order resolution in Isabelle and presented the result at ITP 2016 [37].
- Together with an intern, Jasmin Blanchette, Uwe Waldmann, and Daniel Wand formalized a generalization for the recursive path order and the transfinite Knuth-Bendix order to higher-order terms without λ-abstractions. The result is published in the Isabelle *Archive of Formal Proofs*.

7.1.2. Combination of Satisfiability Procedures

Joint work with Christophe Ringeissen from the PESTO project-team at Inria Nancy – Grand Est, and Paula Chocron at IIIA-CSIC, Bellaterra, Catalonia, Spain.

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).

In 2015, we defined [42] a sound and complete combination procedure à la Nelson-Oppen for the theory of absolutely free data structures (including lists and trees) connected to another theory via bridging functions. This combination procedure has also been refined for standard interpretations. The resulting theory has a nice politeness property, enabling combinations with arbitrary decidable theories of elements. We also investigated [43] other theories amenable to similar combinations: this class includes the theory of equality, the theory of absolutely free data structures, and all the theories in between.

More recently, we have been improving the framework and unified both results. A new paper is in preparation.

7.1.3. Quantifier handling in SMT

Joint work with Andrew J. Reynolds, Univ. of Iowa, USA.

SMT solvers generally rely on various instantiation techniques to handle quantifiers. We are building a unifying framework for handling quantified formulas with equality and uninterpreted functions, such that the major instantiation techniques in SMT solving can be cast in that framework. It is based on the problem of E-ground (dis)unification, a variation of the classic Rigid E-unification problem. We introduced a sound and complete calculus to solve this problem in practice: Congruence Closure with Free Variables (CCFV). Experimental evaluations of implementations of CCFV in the state-of-the-art solver CVC4 and in the solver veriT exhibit improvements in the former and makes the latter competitive with state-of-the-art solvers in several benchmark libraries stemming from verification efforts. A publication is in preparation.

[2] https://bitbucket.org/jasmin_blanchette/isafol/wiki/Home

7.1.4. Non-linear arithmetic in SMT

In the context of the SMArT ANR-DFG (Satisfiability Modulo Arithmetic Theories) and KANASA projects (cf. sections 9.1 and 9.3), we study the theory, design techniques, and implement software to push forward the non-linear arithmetic (NLA) reasoning capabilities in SMT. This year, we designed a framework to combine interval constraint propagation with other decision procedures for NLA, with promising results. We are also currently studying integration of these procedures into combinations of theories. The ideas are validated within the veriT solver, together with code from the raSAT solver (from JAIST). An article is in preparation.

7.1.5. Encoding Set-Theoretic Formulas in First-Order Logic

Proof obligations that arise during the verification of high-level specifications of algorithms in languages such as (Event-)B and TLA$^+$ mix theories corresponding to sets, functions, arithmetic, tuples, and records. Finding encodings of such formulas in the input languages of automatic first-order provers (superposition-based provers or SMT solvers, which are based on multi-sorted first-order logic) is paramount for obtaining satisfactory levels of automation. We describe a method, based on a combination of injection of unsorted expressions into sorted languages, simplification by rewriting, and abstraction, that is the kernel of the SMT backend of the TLA$^+$ proof system (section 6.4). A paper describing our technique was presented at ABZ 2016 [31] and an extension of that article was invited for a special issue of Science of Computer Programming.

During the internship of Matthieu Lequesne, we experimented with an adaptation of the technique for constructing models of formulas in set theory, which could be useful for understanding why proof attempts fail. A prototype generating input for the Nunchaku model finder (section 6.1) allowed us to validate the idea for a core sublanguage of TLA$^+$.

7.1.6. Modal and Description Logics for Graph Transformations

Graph transformations are a research topic that is interesting in its own right, but with many possible applications ranging from the modification of pointer structures in imperative programs, through model transformations in model-driven engineering, to schema-preserving transformations of graph databases. Our particular focus is on verifying these transformations.

Modal logics and variants (such as description logics that are the basis for the web ontology language OWL) have turned out to be suitable specification formalisms because graph structures can typically be perceived as models of modal logics, but these logics suffer from some weaknesses when reasoning about transformations. The first aim of our work was therefore to identify and define sufficiently expressive modal logics, while retaining their pleasant properties, in particular decidability [30].

Our next aim is to implement practically useful proof methods. We have first concentrated on the more natural tableau proofs, with a verification of meta-theoretic properties of the calculi (such as termination) in the Isabelle proof assistant. We now turn to an investigation of encodings as satisfiability problems that can be handled with SAT and SMT solvers, with the hope to achieve a better performance.

7.1.7. Standard Models with Virtual Substitution

Joint work with A. Dolzmann from Leibniz-Zentrum für Informatik in Saarbrücken, Germany.

Extended quantifier elimination for the reals using virtual substitution methods have been successfully applied to various problems in science and engineering. Recently they have attracted attention also as theory solvers within SMT. Such solvers typically ask also for models in the satisfiable case. Models obtained with virtual substitution are in general obtained in certain non-archimedian extension fields of the reals with a corresponding expanded signature. Consequently, the obtained values for the variables include non-standard symbols such as positive infinitesimals and infinite values.

We introduce a complete post-processing procedure to convert our models, for fixed values of parameters, into real models [15]. We furthermore demonstrate the successful application of an implementation of our method within Redlog to a number of extended quantifier elimination problems from the scientific literature including computational geometry, motion planning, bifurcation analysis for models of genetic circuits and for mass action, and sizing of electrical networks. This solves a long-standing problem with the virtual substitution method, which had been explicitly criticized in the scientific literature.

7.1.8. Decidability of Fragments of Free First-Order Logic

We introduce a new decidable fragment of first-order logic with equality, the *Separated Fragment* (SF). It strictly generalizes two already well-known decidable fragments of first-order logic: the Bernays-Schönfinkel-Ramsey (BSR) Fragment and the Monadic Fragment. The defining principle is that universally and existentially quantified variables may not occur together in atoms. Thus, our classification neither rests on restrictions of quantifier prefixes (as in the BSR case) nor on restrictions on the arity of predicate symbols (as in the monadic case).

We show that SF exhibits the finite model property and derive a non-elementary upper bound on the computing time required for deciding satisfiability of SF sentences. For the subfragment of prenex sentences with the quantifier prefix $\exists^*\forall^*\exists^*$ the satisfiability problem is shown to be NEXPTIME-complete. Furthermore, we discuss how automated reasoning procedures can take advantage of our results [34].

Continuing the work presented in the initial publication, we further investigated the computational complexity of SF satisfiability. It nicely scales across the nondeterministic standard complexity classes, depending on joint occurrences of existentially quantified variables from $\exists^*$-blocks that are separated by nonempty $\forall^+$-blocks.

In another line of work, we relaxed the definition of SF, leading to an even larger fragment for which satisfiability is still decidable. In this fragment, variables of $\exists^*$-blocks and $\forall^+$-blocks may occur together in some atom if the respective quantifiers obey a certain order.

7.1.9. Ordered resolution with dismatching constraints

The identification and algorithmic exploration of decidable logic fragments is key to the automation of logics and to obtaining push-button verification technologies. We extend a well-known decidable fragment, linear monadic shallow Horn theories, with straight dismatching constraints, preserving decidability. Furthermore, we show that the restriction to Horn clauses is not needed. The fragment has various applications in security, automata theory and theorem proving [35].

7.1.10. Undecidable combinations of first-order logic with background theories

We show that the universal fragment of Presburger arithmetic augmented with a single uninterpreted predicate (or function) symbol is already undecidable. The result has immediate consequences for verification techniques that combine uninterpreted functions or predicate symbols with (fragments of) Presburger arithmetic. For example, data structures such as arrays can be viewed as a collection of uninterpreted functions that obey certain axioms.

Our result is a sharpening of previously known results. In particular, undecidability holds for a fragment with purely universal quantification: no quantifier alternation is necessary. While in this case the set of unsatisfiable sentences is still recursively enumerable, and in fact hierarchic superposition constitutes a semi-decision procedure, allowing for one quantifier alternation ($\exists\forall$ or $\forall\exists$) leads to a fragment in which neither the satisfiable sentences nor the unsatisfiable ones form a recursively-enumerable set. Hence, there cannot be any refutationally complete calculus for such a combined theory.

7.1.11. Novel techniques for linear arithmetic constraint solving

In [26], [27], we investigate new techniques for linear arithmetic constraint solving. They are based on the linear cube transformation, which allows us to efficiently determine whether a system of linear arithmetic constraints contains a hypercube of a given edge length.

Our first findings based on this transformation are two sound tests that find integer solutions for linear arithmetic constraints. While many complete methods search along the problem surface for a solution, these tests use cubes to explore the interior of the problems. The tests are especially efficient for constraints with a large number of integer solutions, e.g., those with infinite lattice width. Inside the SMT-LIB benchmarks, we have found almost one thousand problem instances with infinite lattice width. Experimental results confirm that our tests are superior on these instances compared to several state-of-the-art SMT solvers.

We also discovered that the linear cube transformation can be used to investigate the equalities implied by a system of linear arithmetic constraints. For this purpose, we developed a method that computes a basis for all implied equalities, i.e., a finite representation of all equalities implied by the linear arithmetic constraints. The equality basis can be used to decide whether a system of linear arithmetic constraints implies a given equality.

7.2. Formal Methods for Developing and Analyzing Algorithms and Systems

Participants: Noran Azmy, Gabriel Corona, Margaux Duroeulx, Marie Duflot-Kremer, Souad Kherroubi, Dominique Méry, Stephan Merz, Nicolas Schnepf, Christoph Weidenbach.

7.2.1. Making explicit domain knowledge in formal system development

Joint work with partners of the IMPEX project.

Modeling languages are concerned with providing techniques and tool support for the design, synthesis and analysis of the models resulting from a given modeling activity, and this activity is usually part of a system development model or process. These languages quite successfully focus on the analysis of the designed system, exploiting the semantic features of the underlying modeling language. These semantics are well understood by the system designers and/or the users of the modeling language, that is why we speak of implicit semantics.

In general, modeling languages are not equipped with resources, concepts or entities handling explicitly domain engineering features and characteristics (domain knowledge) in which the modeled systems evolve.

We posit that designers should explicitly handle the knowledge resulting from an analysis of the application domain, i.e. explicit semantics. As of today, making explicit the domain knowledge inside system design models does not follow any methodological rule; instead, features of domain knowledge are introduced in an ad-hoc way through types, constraints, profiles, etc.

Our claim [11] is that ontologies are good candidates for handling explicit domain knowledge. They define domain theories and provide resources for uniquely identifying concepts of domain knowledge. Therefore, allowing models to make references to ontologies is a modular solution for models to explicitly handle domain knowledge. Overcoming the absence of explicit semantics expression in the modeling languages used to specify systems models will increase the robustness of the designed system models. Indeed, references to the axioms and theorems resulting from the ontologies can be used to strengthen the properties of the designed models. The objective is to offer rigorous mechanisms for handling domain knowledge in design models. We also show how these mechanisms are set up in the cases of formal system models, both for static and dynamic aspects.

7.2.2. Incremental Development of Systems and Algorithms

Joint work with Andriamiarina, Manamiary Bruno, with Neeraj Kumar Singh from IRIT, Toulouse, with Rosemary Monahan, NUI Maynooth, Ireland, and with Zheng Cheng, LINA, Nantes.

The development of distributed algorithms and, more generally, of distributed systems, is a complex, delicate, and challenging process. The approach based on refinement applies a design methodology that starts from the most abstract model and leads, in an incremental way, to a distributed solution. The use of a proof assistant gives a formal guarantee on the conformance of each refinement with the model preceding it.

Our main results during 2016 are:

- An extension [18] for handling the verification of concurrent programs. In a second step, we show the importance of the concept of refinement, and how it can be used to found a methodology for designing concurrent programs using the coordination paradigm.
- A fully mechanized proof [36] of correctness of self-$*$ systems along with an interesting case study related to P2P-based self-healing protocols.

- We report on our progress in implementing a software development environment that integrates two formal software engineering techniques: program refinement as supported by Event-B, and program verification as supported by the Spec# programming system. We improve the usability of formal verification tools by providing a general framework for integrating these two approaches to software verification. We show how the two approaches, based respectively on correctness by construction and on post-hoc verification, can be used in a productive way. In [32], we focus on the final steps in this process where the final concrete specification is transformed into an executable algorithm. We present EB2RC, a plug-in for the RODIN platform that reads in an Event-B model and uses the control framework introduced during its refinement to generate a graphical representation of the executable algorithm. EB2RC also generates a recursive algorithm that is easily translated into executable code. We illustrate our technique through case studies and their analysis.

7.2.3. Verification of the Pastry routing protocol

In her PhD thesis, Noran Azmy develops a formal proof in TLA$^+$ of the routing protocol used in the Pastry protocol [51] for maintaining a distributed hash table over a peer-to-peer network. In a previous thesis [47], Tianxiang Lu had found problems with all published versions of the original protocol, introduced a variant of Pastry, and given a first correctness proof of that protocol, assuming that no node ever disconnects. Due to limitations of TLAPS at that time, Lu's proof relied on many unchecked assumptions on arithmetic and on the underlying data structures, and it was later discovered that several of these assumptions were not valid.

Noran Azmy simplified the proof by introducing intermediate abstractions that allowed her to avoid low-level arithmetic reasoning in the main proof steps, and she proved lemmas corresponding to those assumptions that were actually used in the proof. As a result, she obtained a complete machine-checked proof of Lu's variant of the Pastry protocol, still under the assumption that no node leaves the network. Moreover, a close analysis of the invariant used in her simplified proof revealed that the protocol could be simplified by leaving out the final "lease exchange" protocol. The results were published at ABZ 2016 [22], and an extended article was invited for publication in Science of Computer Programming.

7.2.4. Proof of Determinacy of PharOS

Joint work with Selma Azaiez and Matthieu Lemerre (CEA Saclay), and Damien Doligez (Inria Paris).

As the main contribution of our group to the ADN4SE project funded by PIA, in cooperation with colleagues from CEA LIST, we wrote a high-level TLA$^+$ specification of the real-time operating system PharOS [46] and proved its executions to be deterministic. Roughly speaking, determinacy means that the sequence of local states of any process during a computation does not depend on the order in which processes are scheduled. The proof assumes that no deadlines are missed (which in practice is ensured by schedulability analysis of the particular applications). This property greatly simplifies the analysis and verification of programs that are executed within PharOS. The results were published at ABZ 2016 [21].

7.2.5. Formal Verification of Chains of Security Functions

Joint work with Rémi Badonnel and Abdelkader Lahmadi of the Madynes research group of Inria Nancy.

During his Master's thesis, Nicolas Schnepf studied formal techniques for the automatic verification of chains of security functions in a setting of software-defined networks (SDN). Concretely, he defined an extension of the Pyretic language [44] taking into account the data plane of SDN controllers and implemented a translation of that extension to the input languages of the nuXmv model checker and of SMT solvers. The approach and its scalability was validated over crafted security chains, and a conference paper describing the results is under preparation. Nicolas Schnepf started a PhD thesis in October 2016, jointly supervised by members of the Madynes and VeriDis groups.

7.2.6. Auditing hybrid systems for compliance

There is a huge gap in complexity between the actual analysis of a complex hybrid system and the analysis of the eventual control needed for safe operation. For example, for the combustion process of an engine there is not even a closed formal model, but the eventual control can be represented in a finite domain language.

Such a language can then in particular be used for run-time control of a system through an auditor, providing the detection of faulty components or compliance violations. We have studied the consequences of such an approach if applied to the overall life time process of a technical system [29].

8. Bilateral Contracts and Grants with Industry

8.1. Modeling a Distributed File System

Participant: Stephan Merz.

Our group was contacted by Huawei R&D Silicon Valley for evaluating the suitability of using the TLA$^+$ specification language for describing high-level protocols used in Cloud systems. We provided a specification of protocols used in the Ceph file system [53]. We also provided on-site training for Huawei engineers in Chengdu, China.

8.2. Logic for Business

Participant:

The group in Saarbrücken has established a master agreement with L4B (Logic for Business) on the exchange of data and the creation of bilateral research projects. L4B is involved in several consulting projects with the German car industry on product specification strategies, including software.

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. ANR-DFG Project SMArT

Participants: Haniel Barbosa, Pascal Fontaine, 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 project gathers members of VeriDis in Nancy and Saarbrücken, 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. The results feed back into the implementations of Redlog (section 6.2) and veriT (section 6.5), which also serve as experimentation platforms for theories, techniques and methods designed within this project.

More information on the project can be found on http://smart.gforge.inria.fr/.

9.1.2. ANR Project IMPEX

Participants: Souad Kherroubi, Dominique Méry.

The ANR Project IMPEX, within the INS program, 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. The work reported here also included a cooperation with Pierre Castéran from LaBRI Bordeaux.

Modeling languages provide techniques and tool support for the design, synthesis, and analysis of the models resulting from a given modeling activity, as part of a system development process. These languages quite successfully focused on the analysis of the designed system exploiting the expressed semantic power of the underlying modeling language. The semantics of this modeling languages are well understood by the system designers and the users of the modeling language, i.e. the semantics is implicit in the model. In general, modeling languages are not equipped with resources, concepts or entities handling explicitly domain engineering features and characteristics (domain knowledge) underlying the modeled systems. Indeed, the designer has to explicitly handle the knowledge resulting from an analysis of this application domain [49], i.e. explicit semantics. Nowadays, making explicit the domain knowledge inside system design models does not obey any methodological rules validated by practice. The users of modeling languages introduce these domain knowledge features through types, constraints, profiles, etc. Our claim is that ontologies are good candidates for handling explicit domain knowledge. They define domain theories and provide resources for uniquely identifying domain knowledge concepts. Therefore, allowing models to make references to ontologies is a modular solution for models to explicitly handle domain knowledge. Overcoming the absence of explicit semantics expression in the modeling languages used to specify systems models will increase the robustness of the designed system models. Indeed, the axioms and theorems resulting from the ontologies can be used to strengthen the properties of the designed models. The objective [11] is to offer rigorous mechanisms for handling domain knowledge in design models.

9.1.3. Inria IPL HAC SPECIS

Participants: Marie Duflot-Kremer, Stephan Merz.

The goal of the HAC SPECIS (High-performance Application and Computers: Studying PErformance and Correctness In Simulation) project is to answer methodological needs of HPC application and runtime developers and to allow studying real HPC systems with respect to both correctness and performance. To this end, this Inria Project Lab assembles experts from the HPC, formal verification, and performance evaluation communities.

HAC SPECIS started in 2016. VeriDis contributes through its expertise in formal verification techniques. In particular, our goal is to extend the functionalities of exhaustive and statistical model checking within the SimGrid platform.

9.1.4. Inria Technological Development Action CUIC

Participants: Jasmin Christian Blanchette, Simon Cruanes.

Most “theorems” initially given to a proof assistant are incorrect, whether because of a typo, a missing assumption, or a fundamental flaw. Novices and experts alike can enter invalid formulas and find themselves wasting hours, or even days, on an impossible proof. This project, funded by Inria and running from 2015 to 2017, supports the development of a counterexample generator for higher-order logic. This new tool, called Nunchaku (cf. section 6.1), will be integrated in various proof assistants, including Isabelle, Coq, and the TLA$^+$ Proof System. The project is coordinated by Jasmin Blanchette and also involves Inria Saclay (Toccata group) and Inria Rennes (Celtique group), among others. Simon Cruanes was hired in October 2015 and has started the development of Nunchaku, whereas Blanchette has developed an Isabelle frontend. Three releases have taken place so far, and the tool is an integral part of the Isabelle2016-1 official release. Work has started on Coq and TLAPS frontends. The tool is described in a conference publication [33] and was presented at a workshop [28].

9.1.5. Inria ADT PLM (2014-2016)

Participant: Matthieu Nicolas.

Joint work with Gérald Oster (project-team Coast, Inria Nancy – Grand Est) and Martin Quinson (project-team Myriads, Inria Rennes – Bretagne Atlantique)

The goal of this project is to establish an experimental platform for studying the didactics of informatics, specifically centered on introductory programming courses.

The project builds upon a pedagogical platform for supervising programming exercises developed for our own teaching, and improves this base in several ways. We want to provide more adapted feedback to the learners, and gather more data to better understand how beginners learn programming.

This year, we finalized the web version of our framework, and submitted several project applications to pursue this work in the future. Unfortunately, none of these applications have been accepted so far. Martin Quinson invited Peter Hubwieser, professor of the Technical University of Munich (TUM) and specialist of the didactics of Computer Science, for two weeks in November. Developing the PLM and exploiting the data already gathered were central elements of this work meeting. A joint publication is currently prepared, targeting the ItiCSE'17 conference.

9.2. European Initiatives

9.2.1. FP7 & H2020 Projects

Program: H2020-FETOPEN-2015-CSA

Project acronym: SC^2

Project title: Satisfiability Checking and Symbolic Computation

Duration: July 2016 – September 2018

Coordinator: James H. Davenport (U. Bath, U.K.)

Other partners: RWTH Aachen (Germany), Fondazione Bruno Kessler (Italy), Università degli Studi di Genova (Italy), Maplesoft Europe Ltd (Germany), Coventry University (U.K.), University of Oxford (U.K.), Universität Kassel (Germany), Max Planck Institut für Informatik (Germany), Universität Linz (Austria)

Abstract: Whereas symbolic computation is concerned with efficient algorithms for determining exact solutions to complex mathematical problems, more recent developments in the area of satisfiability checking tackle similar problems with different algorithmic and technological solutions. Both communities have made remarkable progress in the last decades and address practical problems of rapidly increasing size and complexity. For example, satisfiability checking is an essential backend for assuring the security and the safety of computer systems. Techniques and tools of symbolic computation are used by different scientific communities for solving large mathematical problems that are out of reach of pencil and paper developments. Currently the two communities are largely disjoint and unaware of the achievements of each other, despite strong reasons for them to discuss and collaborate, as they share many central interests. Bridges between the communities in the form of common platforms and roadmaps are necessary to initiate an exchange, and to support and to direct their interaction. This Coordination and Support Action within the FET-Open framework will initiate a wide range of activities to bring the two communities together, identify common challenges, offer global events and bilateral visits, propose standards, and so on. Combining the knowledge, experience and the technologies in these communities will lead to cross-fertilization and mutual improvements, enabling the development of radically improved software tools.

9.3. International Initiatives

9.3.1. Inria International Partners

9.3.1.1. KANASA

Title: Kanazawa-Nancy for Satistifiability and Arithmetics

International Partner: Japan Advanced Institute for Science and Technology (Dept. Intelligent Robotics, Mizuhito Ogawa)

Starting year: 2016

During the last decade, there has been tremendous progress on symbolic verification techniques, spurred in particular by the development of SMT (satisfiability modulo theories) techniques and tools. Our first direction of research will be to investigate the theoretical background and the practical techniques to integrate Interval Constraint Propagation within a generic SMT framework, including other decision procedures and quantifier handling techniques. On the purely arithmetic side, we also want to study how to unite the reasoning power of all arithmetic techniques developed in the team, including simplex-based SMT-like reasoners, Virtual Substitution, and Cylindrical Algebraic Decomposition. In particular, this includes developing theory combination frameworks for linear and non-linear arithmetic. There is a strong incentive for these kind of combinations since even non-linear SMT problems contain a large proportion of linear constraints. The partnership is supported by a Memorandum of Understanding between JAIST and LORIA.

9.4. International Research Visitors

9.4.1. Visits of International Scientists

Ilina Stoilkovska

Date: 1 September – 31 October

Institution: TU Wien (Austria)

Host: Stephan Merz

Ilina is a PhD student at TU Wien, Austria, and works on tailored abstractions for the parameterized verification of fault-tolerant distributed algorithms. During her stay in Nancy, she worked on a formal soundness proof of her abstractions in the TLA$^+$ Proof System.

Tung Vu Xuan

Date: 1 May 2016 – 30 April 2017

Institution: JAIST

Host: Pascal Fontaine

Tung Vu Xuan is a PhD student at JAIST, Japan. He is visiting VeriDis in the context of the KANASA project. He works mainly on Interval Constraint Propagation (ICP), a heuristic but powerful method for satisfiability checking of non-linear arithmetic (NLA) constraints. During his stay, we investigate techniques to combine ICP with decision procedures for NLA within an SMT context.

9.4.2. Internships

Anders Olav Candasamy

Date: 1 March – 31 July

Institution: Université de Lorraine (Erasmus Mundus DESEM)

Host: Dominique Méry

Anders Candasamy analyzed a hemodialysis case study using Event-B. Besides developing the formal model, he also reflected on the modeling process and proposed several methodological improvements.

Matthieu Lequesne

Date: 1 March – 31 July

Institution: École Polytechnique

Host: Stephan Merz

Matthieu Lequesne worked on translating formulas in a core sublanguage of TLA$^+$ to the input format of Nunchaku (section 6.1), with the aim of producing (counter)models for TLA$^+$ proof obligations.

Weichung Shaw

Date: 1 March – 31 August

Institution: Université de Lorraine (Erasmus Mundus DESEM)

Host: Stephan Merz

Weichung Shaw worked on formalizing a correctness proof of the Raft consensus algorithm [50] in TLA$^+$. He proved several fundamental lemmas and documented several methodological issues with the use of TLAPS.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Organization of Scientific Events

10.1.1.1. General Chair, Scientific Chair

Jasmin Blanchette and Stephan Merz, with the help of Anne-Lise Charbonnier of Inria Nancy, organized the *7th International Conference on Interactive Theorem Proving* (ITP 2016) and associated workshops in Nancy, on August 22–27, 2016.

10.1.1.2. Member of the Organizing Committees

Jasmin Blanchette co-organized the *Hammers for Type Theories* (HaTT 2016) workshop at IJCAR 2016 in Coimbra, Portugal.

Pascal Fontaine co-organized the *First SC*^{2} *workshop on Satisfiability Checking and Symbolic Computation* with Erika Abraham (RWTH, Aachen).

Pascal Fontaine co-organized the 5th Workshop on Practical Aspects of Automated Reasoning (PAAR) with Stephan Schulz (DHBW Stuttgart) and Josef Urban (Czech Technical University in Prague).

Dominique Méry was a member of the organizing committees of the workshops F-IDE, BWare, Impex, and Formose.

Dominique Méry, together with Yamine Aït-Ameur (Toulouse) and Shin Nakajima (Tokyo), organized a meeting on *Implicit and explicit semantics integration in proof based developments of discrete systems* in November within the series of NII Shonan meetings.

The International Summer School on Verification Techniques, Systems, and Applications (VTSA) has been organized since 2008 in the Greater Region (Nancy, Saarbrücken, Luxembourg, Liège, and Koblenz), and Stephan Merz and Christoph Weidenbach are co-organizers of VTSA. In 2016, VTSA took place at the end of August in Liège, Belgium.

10.1.2. Selection of Scientific Events

10.1.2.1. Chair of Conference Program Committees

Jasmin Blanchette and Stephan Merz chaired the program committee of the *7th International Conference on Interactive Theorem Proving* (ITP 2016).

Stephan Merz co-chaired the program committee of the Third International Workshop on Formal Reasoning in Distributed Algorithms (FRiDA), organized in May as a satellite of NETYS in Marrakech, Morocco.

10.1.2.2. Member of Conference Program Committees

Jasmin Blanchette served on the program committee of the *International Conference on Tests and Proofs* (TAP).

Pascal Fontaine served on the program committee of the workshop SMT.

Stephan Merz served on the program committees of the international conferences ABZ, ICALP, and ICFEM, and of the workshops ARQNL, FMICS-AVoCS, and GRSRD.

Martin Strecker served on the program committees of ICTERI and ICGT.

Thomas Sturm served on the program committees of CASC and of the SC^2 workshop at SYNACS.

Uwe Waldmann served on the program committee of the workshop PAAR, colocated with IJCAR 2016.

Christoph Weidenbach served on the program committee of IJCAR.

10.1.3. Journals

Stephan Merz, together with Jun Pang of the University of Luxembourg, edited two volumes of a special issue on Formal Engineering Methods in the journal *Formal Aspects of Computing*.

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. Together with Deepak Kapur and Stéphane Demri he edited a special issue of JAR containing selected and extended papers of IJCAR 2014.

10.1.4. Invited Talks

Jasmin Blanchette gave invited talks at the Semantic Representation of Mathematical Knowledge Workshop organized by the Wolfram Foundation and the Fields Institute in Toronto, Canada, at the Sino-German Frontiers of Science Symposium (SINOGFOS) organized by the Humboldt Foundation and the Chinese Academy of Science in Shenzhen, China, at the Workshop on Proofs, Justifications, and Certificates in Toulouse, France, at the Universality of Proof seminar at Schloss Dagstuhl in Wadern, Germany, and at the Prague Inter-Reasonning Workshop (PIWo) in Prague, Czech Republic.

Pascal Fontaine gave an invited talk at the AFSEC day of the GdR GPL, and at GT-Verif day of the GdR IM.

Stephan Merz gave invited talks at the TRS meeting and the JAIST-LORIA workshop in Kanazawa, Japan, on *Satisfiability Checking for Modal Logics via SMT Solving* and on *The Design of the TLA$^+$ Proof System*. He also gave an invited talk at the *Cloud Reliability Workshop* in Shenzhen, China, on *A Formal Analysis of Pastry*.

Thomas Sturm gave an invited talk at ACA 2016 titled *Real Problems over the Reals*.

Christoph Weidenbach gave an invited lecture at the SMT Summer School in Lisbon, Portugal.

10.1.5. Leadership within the Scientific Community

Jasmin Blanchette served as editor of the newsletter of the Association for Automated Reasoning (AAR) and as member of the AAR board.

Jasmin Blanchette and Christoph Weidenbach were elected on the CADE (*Conference on Automated Deduction*) Inc. Board of Trustees. Christoph Weidenbach was elected President of CADE Inc. by the CADE Inc. Board of Trustees.

Jasmin Blanchette is an ex officio member of the steering committee of the conference series *Interactive Theorem Proving*.

Pascal Fontaine is an SMT-LIB manager, together with Clark Barrett (Stanford University) and Cesare Tinelli (University of Iowa). He is a member of the FroCoS steering Committee. He has been an elected CADE trustee since October 2014. He serves as member of the Association for Automated Reasoning (AAR) board.

Stephan Merz is a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*. He is also a member of the steering committee of the workshop on Automated Verification of Critical Systems (AVoCS).

Thomas Sturm is a member of the steering committee of the conference series *Mathematical Aspects of Computer and Information Sciences* (MACIS).

Christoph Weidenbach is a member of the steering committee of IJCAR.

10.1.6. Scientific Expertise

Pascal Fontaine was a panel member for the CASC-25 competition of first-order theorem prover.

Stephan Merz served as an expert for the French Agence Nationale de la Recherche (ANR), the Haut Conseil de l'Évaluation de la Recherche et de l'Enseignement Supérieur (HCERES), and for the European Research Council (ERC).

Christoph Weidenbach served as an expert for GIF (German Israel Foundation), the FWF (Austrian Science Fund) and the DFG (German Science Foundation).

10.1.7. Research Administration

Dominique Méry is the head of the Doctoral School IAEM Lorraine for the University of Lorraine.

Stephan Merz is a member of the Scientific Directorate of the International Computer Science Meeting Center in Schloss Dagstuhl. Until August 2016, he was the head of the PhD committee for computer science of the Doctoral School IAEM Lorraine. Since September 2016, he is the delegate for scientific affairs at the Inria Nancy – Grand Est research center. He is also the delegate for the organization of conferences at Inria Nancy and the coordinator of the CPER *Sciences du Numérique* in Lorraine (2015–2020). He was a member of the hiring committee of junior researchers at Inria Nancy in 2016 and a member of the committee for the SIF thesis award (*Prix Gilles Kahn*).

Christoph Weidenbach is a member of the selection committee of the Saarbrücken Graduate School in Computer Science.

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

Master: Jasmin Blanchette, Computational Metaphysics (guest lecturer), 4 HETD, Freie Universität Berlin, Germany.

Licence: Marie Duflot-Kremer, Algorithmique et Programmation 1, 80 HETD L1 Mathématiques, Informatiques Sciences pour l'Ingénieur, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Bases de données 1 et 2, 35 HETD, L2 informatique, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Projet personnel et communication, 50 HETD, L2 informatique, Université de Lorraine, France.

Master : Marie Duflot-Kremer, Vérification de systèmes, 30 HETD, M1 Informatique, Université de Lorraine, France.

Master: Marie Duflot-Kremer and Stephan Merz, Vérification algorithmique, 30 HETD, M2 Informatique, Université de Lorraine, France.

Master: Marie Duflot-Kremer and Stephan Merz, Elements of Model Checking, 36 HETD, M2 Informatique and Master Erasmus Mundus DESEM, Université de Lorraine, France.

Master : Marie Duflot-Kremer and Stephan Merz, Conception et architectures distribuées 24 HETD M1 informatique, Université de Lorraine

Licence : Pascal Fontaine, Structure des ordinateurs, 67 HETD, L2 MIASHS, parcours MIAGE, Université de Lorraine, France.

Licence : Pascal Fontaine, Logique des prédicats, 32 HETD, L2 MIASHS, Université de Lorraine, France.

Master : Pascal Fontaine, Réseaux, 50 HETD, M1 MIAGE, Université de Lorraine, France.

Master : Pascal Fontaine, Génie Logiciel, 30 HETD, M1 MIAGE, IGA Rabbat et Université de Lorraine, Maroc.

Master: Dominique Méry, Models and algorithms, 60 HETD, M1, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Formal model engineering, 24 HETD, M2, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Modeling Systems, 30 HETD, M2, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Modeling Systems, 36 HETD, M2 informatique and Master Erasmus Mundus DESEM, Université de Lorraine, France.

Master: Dominique Méry, Event-B modeling, 8 HETD, NUI Maynooth.

Master: Uwe Waldmann, Automated Reasoning I, 90 HETD, Universität des Saarlandes, Germany.

Master: Uwe Waldmann, Automated Reasoning II, 60 HETD, Universität des Saarlandes, Germany. This lecture received the teaching award of the Computer Science Students Association.

10.2.2. Supervision

PhD: Noran Azmy, An Automated Proof of Correctness for Pastry, Saarland University and Université de Lorraine, defended on November 24, 2016.

PhD: Marek Košta, Computational Logic, Universität des Saarlandes. Defended on December 13, 2016.

PhD in progress: Gabor Alági, Efficient Reasoning in Finite Domains, Saarland University. Supervised by 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: Mathias Fleury, Formalization of Logical Calculi, Saarland University. Supervised by Christoph Weidenbach and Jasmin Blanchette, since 09/2015.

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 and Jasmin Blanchette, since 02/2011.

10.2.3. Thesis committees

Dominique Méry served on the committees for the PhD thesis of Pierre Halmagrand (CNAM) and the habilitation thesis of Brahim Hamid (Université Toulouse Jean Jaurès).

Stephan Merz served as a reviewer for the PhD thesis of Yakoub Némouchi (Université Paris Saclay) and as a PhD examiner for the PhD thesis of Alland Blanchard (Université d'Orléans).

10.3. Science outreach

Marie Duflot-Kremer took part in various science outreach activities, with a public ranging from primary school kids to golden agers, including high school and potential university students. A selection of these activities is given below:

- two days at "Fête de la science" in Nancy (Faculté de Sciences et Technologies and ARTEM);

- a course on Scratch for high school professors in charge of teaching optional course ISN (Informatique et Sciences du Numérique);
- her explanations of three new unplugged activities (data bases, model checking and text compression) have been recorded by Inria and will soon be added to the Youtube channel of Interstice intended for promoting and sharing such activities;
- she is in charge of the scientific part of the second module in the Class'Code project, aiming at training teachers and educators for carrying out computer science activities with childrens aged 8 to 14 years;
- she is a member of two groups including university and secondary school teachers, dedicated to the training of math teachers who now teach computer science to students of age 11 to 18. A day of training was given to high school teachers;
- "Journée femmes de Sciences": one day dedicated to the promotion of science towards 14 year-old girls;
- she is a member of the steering committee preparing an itinerant exposition intended for explaining computer science to the public, to be released in December 2016;
- she presented unplugged outreach activities to the staff at Cité des Sciences (Paris);
- she conducted during five months an experiment on the discovery of programming for golden agers using Scratch;
- she took part in "Pépinière 4.0" and 4.1, explaining computer science concepts to teachers.

11. 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

Articles in International Peer-Reviewed Journals

[11] Y. AIT AMEUR, D. MÉRY. *Making explicit domain knowledge in formal system development*, in "Science of Computer Programming", March 2016, vol. 121, n^o 100–127 [*DOI :* 10.1016/J.SCICO.2015.12.004], https://hal.inria.fr/hal-01245832

[12] J. C. BLANCHETTE, S. BÖHME, M. FLEURY, S. J. SMOLKA, A. STECKERMEIER. *Semi-intelligible Isar Proofs from Machine-Generated Proofs*, in "Journal of Automated Reasoning", 2016 [*DOI :* 10.1007/S10817-015-9335-3], https://hal.inria.fr/hal-01211748

[13] J. C. BLANCHETTE, D. GREENAWAY, C. KALISZYK, D. KÜHLWEIN, J. URBAN. *A Learning-Based Fact Selector for Isabelle/HOL*, in "Journal of Automated Reasoning", 2016, vol. 57, pp. 219 - 244 [*DOI :* 10.1007/S10817-016-9362-8], https://hal.inria.fr/hal-01386986

[14] J. C. BLANCHETTE, C. KALISZYK, L. C. PAULSON, J. URBAN. *Hammering towards QED*, in "Journal of Formalized Reasoning", 2016, vol. 9, n^o 1, pp. 101-148, https://hal.inria.fr/hal-01386988

[15] M. KOŠTA, T. STURM, A. DOLZMANN. *Better answers to real questions*, in "Journal of Symbolic Computation", 2016, vol. 74, pp. 255 - 275 [*DOI :* 10.1016/J.JSC.2015.07.002], https://hal.inria.fr/hal-01388720

[16] S. MERZ, J. PANG. *Editorial*, in "Formal Aspects of Computing", 2016, vol. 28, n^o 3, pp. 343-344 [*DOI :* 10.1007/S00165-016-0390-2], https://hal.inria.fr/hal-01356470

[17] S. MERZ, J. PANG. *Editorial*, in "Formal Aspects of Computing", 2016, vol. 28, n^o 5, pp. 723-724 [*DOI :* 10.1007/S00165-016-0390-2], https://hal.inria.fr/hal-01356471

[18] D. MÉRY. *Playing with State-Based Models for Designing Better Algorithms*, in "Future Generation Computer Systems", May 2016, 25 p. , https://hal.inria.fr/hal-01316026

Invited Conferences

[19] *Best Paper*
A. REYNOLDS, J. C. BLANCHETTE. *A Decision Procedure for (Co)datatypes in SMT Solvers*, in "IJCAI 2016", New York City, United States, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, July 2016, https://hal.inria.fr/hal-01397082.

International Conferences with Proceedings

[20] E. H. ABRAHÁM, J. ABBOTT, B. BECKER, A. M. BIGATTI, M. M. BRAIN, B. BUCHBERGER, A. CIMATTI, J. H. DAVENPORT, M. M. ENGLAND, P. FONTAINE, S. M. FORREST, A. GRIGGIO, D. KROENING, W. M. SEILER, T. STURM. *SC 2 : Satisfiability Checking meets Symbolic Computation (Project Paper)*, in "Intelligent Computer Mathematics", Bialystok, Poland, July 2016, https://hal.inria.fr/hal-01377655

[21] S. AZAIEZ, D. DOLIGEZ, M. LEMERRE, T. LIBAL, S. MERZ. *Proving Determinacy of the PharOS Real-Time Operating System*, in "Abstract State Machines, Alloy, B, TLA, VDM, and Z - 5th International Conference, ABZ 2016", Linz, Austria, M. J. BUTLER, K.-D. SCHEWE, A. MASHKOOR, M. BIRÓ (editors), LNCS - Lecture Notes in Computer Science, Springer, May 2016, vol. 9675, pp. 70-85 [*DOI :* 10.1007/978-3-319-33600-8_4], https://hal.inria.fr/hal-01322335

[22] N. AZMY, S. MERZ, C. WEIDENBACH. *A Rigorous Correctness Proof for Pastry*, in "Abstract State Machines, Alloy, B, TLA, VDM, and Z - 5th International Conference, ABZ 2016", Linz, Austria, M. J. BUTLER, K.-D. SCHEWE, A. MASHKOOR, M. BIRÓ (editors), Springer, 2016, vol. 9675, pp. 86-101 [*DOI :* 10.1007/978-3-319-33600-8_5], https://hal.inria.fr/hal-01322342

[23] J. C. BLANCHETTE, A. BOUZY, A. LOCHBIHLER, A. POPESCU, D. TRAYTEL. *Friends with Benefits: Implementing Foundational Corecursion in Isabelle/HOL (Extended Abstract)*, in "Isabelle Workshop 2016", Nancy, France, August 2016, https://hal.inria.fr/hal-01401812

[24] J. C. BLANCHETTE, M. FLEURY, C. WEIDENBACH. *A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality (Extended Abstract)*, in "Isabellle Workshop 2016", Nancy, France, August 2016, https://hal.inria.fr/hal-01401807

[25] *Best Paper*
J. C. BLANCHETTE, M. FLEURY, C. WEIDENBACH. *A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality*, in "8th International Joint Conference on Automated Reasoning (IJCAR 2016)", Coimbra, Portugal, Automated Reasoning - 8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, June 27 - July 2, 2016, Proceedings, June 2016 [*DOI :* 10.1007/978-3-319-40229-1_4], https://hal.inria.fr/hal-01336074.

[26] M. BROMBERGER, C. WEIDENBACH. *Computing a Complete Basis for Equalities Implied by a System of LRA Constraints*, in "14th International Workshop on Satisfiability Modulo Theories", Coimbra, Portugal, T. KING, R. PISKAC (editors), CEUR Workshop Proceedings, 2016, vol. 1617, pp. 15-30, https://hal.inria.fr/hal-01403214

[27] M. BROMBERGER, C. WEIDENBACH. *Fast Cube Tests for LIA Constraint Solving*, in "Automated Reasoning - 8th International Joint Conference (IJCAR 2016)", Coimbra, Portugal, N. OLIVETTI, A. TIWARI (editors),

Lecture Notes in Computer Science, Springer, 2016, vol. 9706, pp. 116-132 [*DOI :* 10.1007/978-3-319-40229-1_9], https://hal.inria.fr/hal-01403200

[28] S. CRUANES, J. C. BLANCHETTE. *Extending Nunchaku to Dependent Type Theory*, in "Hammers for Type Theories (HaTT 2016)", Coimbra, Portugal, Proceedings First International Workshop on Hammers for Type Theories, July 2016, vol. 210, pp. 3 - 12 [*DOI :* 10.4204/EPTCS.210.3], https://hal.inria.fr/hal-01401696

[29] C. FETZER, C. WEIDENBACH, P. WISCHNEWSKI. *Compliance, Functional Safety and Fault Detection by Formal Methods*, in "Leveraging Applications of Formal Methods, Verification and Validation (ISOLA 2016)", Corfu, Greece, T. MARGARIA, B. STEFFEN (editors), Lecture Notes in Computer Science, Springer, 2016, vol. 9953, pp. 626 - 632 [*DOI :* 10.1007/978-3-319-47169-3_48], https://hal.inria.fr/hal-01403190

[30] J. HAËL BRENAS, R. ECHAHED, M. STRECKER. *Ensuring Correctness of Model Transformations While Remaining Decidable*, in "Theoretical Aspects of Computing - ICTAC", Taipei, Taiwan, Theoretical Aspects of Computing – ICTAC 2016 13th International Colloquium, Taipei, Taiwan, ROC, October 24–31, 2016, Proceedings, October 2016, pp. 315 - 332 [*DOI :* 10.1007/978-3-319-46750-4_18], https://hal.archives-ouvertes.fr/hal-01403585

[31] S. MERZ, H. VANZETTO. *Encoding TLA+ into Many-Sorted First-Order Logic*, in "Abstract State Machines, Alloy, B, TLA, VDM, and Z - 5th International Conference, ABZ 2016", Linz, Austria, M. J. BUTLER, K.-D. SCHEWE, A. MASHKOOR, M. BIRÓ (editors), Springer, 2016, vol. 9675, pp. 54-69 [*DOI :* 10.1007/978-3-319-33600-8_3], https://hal.inria.fr/hal-01322328

[32] D. MÉRY, R. MONAHAN, C. ZHENG. *On two Friends for getting Correct ProgramsAutomatically Translating Event B Specifications to Recursive Algorithms in Rodin*, in "ISOLA 2016", CORFU, Greece, B. STEFFEN, T. MARGARIA (editors), Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques, Springer, October 2016, vol. I, n^{o} 9952, 18 p. [*DOI :* 10.1007/978-3-319-47166-2_57], https://hal.inria.fr/hal-01369425

[33] A. REYNOLDS, J. C. BLANCHETTE, S. CRUANES, C. TINELLI. *Model Finding for Recursive Functions in SMT*, in "8th International Joint Conference on Automated Reasoning (IJCAR 2016)", Coimbra, Portugal, Automated Reasoning - 8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, June 27 - July 2, 2016, Proceedings, June 2016 [*DOI :* 10.1007/978-3-319-40229-1_10], https://hal.inria.fr/hal-01336082

[34] T. STURM, M. VOIGT, C. WEIDENBACH. *Deciding First-Order Satisfiability when Universal and Existential Variables are Separated*, in "LICS 2016", New York, United States, July 2016, pp. 86 - 95 [*DOI :* 10.1145/2933575.2934532], https://hal.inria.fr/hal-01389744

[35] A. TEUCKE, C. WEIDENBACH. *Ordered Resolution with Straight Dismatching Constraints*, in "5th Workshop on Practical Aspects of Automated Reasoning (PAAR 2016)", Coimbra, Portugal, P. FONTAINE, S. SCHULZ, J. URBAN (editors), CEUR Workshop Proceedings, 2016, vol. 1635, pp. 95-109, https://hal.inria.fr/hal-01403206

Scientific Books (or Scientific Book chapters)

[36] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Incremental Proof-Based Development for Resilient Distributed Systems*, in "Trustworthy Cyber-Physical Systems Engineering", Trustworthy Cyber-Physical Systems Engineering, Taylor and Francis Group, September 2016, https://hal.archives-ouvertes.fr/hal-01246669

Books or Proceedings Editing

[37] J. C. BLANCHETTE, S. MERZ (editors). *Interactive Theorem Proving: 7th International Conference, ITP 2016*, Lecture Notes in Computer Science, Springer, Nancy, France, 2016, vol. 9807 [*DOI :* 10.1007/978-3-319-43144-4], https://hal.inria.fr/hal-01356464

References in notes

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

[39] 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

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

[41] 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

[42] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *A Polite Non-Disjoint Combination Method: Theories with Bridging Functions Revisited*, in "25th International Conference on Automated Deduction, CADE-25", Berlin, Germany, A. P. FELTY, A. MIDDELDORP (editors), Lecture Notes in Computer Science, Springer, August 2015, vol. 9195, pp. 419-433 [*DOI :* 10.1007/978-3-319-21401-6_29], https://hal.inria.fr/hal-01157898

[43] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *A Rewriting Approach to the Combination of Data Structures with Bridging Theories*, in "Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015", Wroclaw, Poland, C. LUTZ, S. RANISE (editors), Lecture Notes in Computer Science, Springer, September 2015, vol. 9322, pp. 275–290 [*DOI :* 10.1007/978-3-319-24246-0_17], https://hal.inria.fr/hal-01206187

[44] N. FOSTER, A. GUHA, M. REITBLATT, A. STORY, M. J. FREEDMAN, N. PRAVEEN KATTA, C. MONSANTO, J. REICH, J. REXFORD, C. SCHLESINGER, D. WALKER, R. HARRISON. *Languages for software-defined networks*, in "IEEE Communications Magazine", 2013, vol. 51, n^{o} 2, pp. 128-134

[45] L. LAMPORT. *Specifying Systems*, Addison-Wesley, Boston, Mass., 2002

[46] M. LEMERRE, E. OHAYON. *A Model of Parallel Deterministic Real-Time Computation*, in "Proc. 33rd IEEE Real-Time Systems Symposium (RTSS 2012)", San Juan, PR, U.S.A., IEEE Comp. Soc., 2012, pp. 273-282

[47] T. LU. *Formal Verification of the Pastry Protocol*, Universität des Saarlandes and Université de Lorraine, 2013

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

[49] D. MÉRY, S. RUSHIKESH, A. TARASYUK. *Integrating Domain-Based Features into Event-B: a Nose Gear Velocity Case Study*, in "Model and Data Engineering - 5th International Conference, MEDI 2015", Rhodos, Greece, L. BELLATRECHE, Y. MANOLOPOULOS (editors), LNCS, Springer, 2015, vol. 9344, pp. 89-102, https://hal.inria.fr/hal-01245991

[50] D. ONGARO, J. K. OUSTERHOUT. *In Search of an Understandable Consensus Algorithm*, in "USENIX Annual Technical Conference 2014", Philadelphia, PA, G. GIBSON, N. ZELDOVICH (editors), Usenix Association, 2014, pp. 305-319

[51] A. ROWSTRON, P. DRUSCHEL. *Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems*, in "IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)", Heidelberg, Germany, R. GUERRAOUI (editor), Lecture Notes in Computer Science, Springer, 2001, vol. 2218, pp. 329-350

[52] 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

[53] S. A. WEIL, S. A. BRANDT, E. L. MILLER, D. D. E. LONG, C. MALTZAHN. *Ceph: A Scalable, High-Performance Distributed File System*, in "7th Symp. Operating Systems Design and Implementation (OSDI '06)", Seattle, WA, Usenix Association, 2006, pp. 307-320