INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team VeriDis

VERIfication of DIstributed Systems

Nancy - Grand Est

Theme : Programs, Verification and Proofs

Table of contents

1. Team

Research Scientist

Stephan Merz [Team leader, Senior Researcher Inria, HdR]

Faculty Members

Pascal Fontaine [Associate Professor, University Nancy 2]
Dominique Méry [Professor, University Henri Poincaré Nancy 1, HdR]

External Collaborator

David Déharbe [Universidade Federal do Rio Grande de Norte, Brazil]

Technical Staff

Thomas Bouton [until 08/2010, funded by ANR project Decert]

PhD Students

Sabina Akhtar [University Henri Poincaré Nancy 1, since 09/2008, grant from foreign government]
Diego Caminha Barbosa de Oliveira [University Nancy 2, since 12/2007, INRIA grant]
Henri Debrat [University Henri Poincaré Nancy 1, since 10/2009, ministery grant, joint supervision with Bernadette Charron-Bost]
Tianxiang Lu [Universität des Saarlands, since 05/2009, joint supervision with Christoph Weidenbach]
Cristián Rosa [University Henri Poincaré Nancy 1, since 12/2008, joint supervision with Martin Quinson of AlGorille team, funded by ANR project UMSSimGrid]
Hernán-Pablo Vanzetto [University Henri Poincaré Nancy 1, since 10/2010, funded by MSR-INRIA Centre, joint supervision with Kaustuv Chaudhuri]

Post-Doctoral Fellow

Bruno Woltzenlogel Paleo

Administrative Assistant

Emmanuelle Deschamps [shared with teams Caramel, Cassis, Orpailleur]

2. Overall Objectives

2.1. Introduction

VeriDis was created in January 2010 as a local team of INRIA Nancy-Grand Est. It is currently under evaluation for the creation as an INRIA project team that would include, beyond its current members, participants from the Automation of Logic Research Group at Max-Planck Institut für Informatik in Saarbrücken, led by Christoph Weidenbach. The VeriDis members in Nancy belong to the MOSEL research group of LORIA, the computer science research laboratory in Nancy. MOSEL is a joint team of CNRS and of Nancy University.

The objective of VeriDis is to exploit and further develop the advances and integration of interactive and automated theorem proving, with applications to the area of concurrent and distributed systems. The goal of our project is to assist algorithm and system designers to carry out formally proved developments, where proofs of relevant properties, as well as bugs, can be found with a high degree of automation.

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

VeriDis proposes to exploit and further develop automation in system verification, and to apply its techniques within the context of concurrent and distributed algorithms, which are by now ubiquitous and whose verification is a big challenge. Concurrency problems are central to the development and verification of programs for multi- and many-core architectures, and distributed computation underlies the paradigms of grid and cloud computing. Typical problems that we intend to address include the verification of algorithms and protocols for peer-to-peer and overlay networks, such as distributed hash tables, multicast trees or gossip-based protocols. The added resilience to component failures gained by distributed computation is one of the motivations for its adoption, and constitutes another challenge for verification. On success of VeriDis, automated, full-fledged verification will be available for significant properties of today's concurrent and distributed algorithms. We aim to move current research in this area on to a new level of productivity and quality. To give a concrete example: today a network protocol engineer designing a new distributed protocol may validate it using testing or model checking. Model checking will help finding bugs, but can only guarantee properties of a high-level model of the protocol, usually restricted to finite instances. Testing distributed systems and protocols is notoriously difficult because corner cases are hard to establish and reproduce. Also, many testing techniques require implementation, which is expensive and time-consuming, and errors are found only when they can no longer be fixed cheaply. The techniques that we develop aim at automatically proving significant properties of the protocol already at the design phase. Our methods will be applicable to designs and algorithms that are typical for components of operating systems, distributed services, and down to the (mobile) network systems industry.

2.2. Highlights of the year

- This year, the veriT solver (see section 5.1) improved significantly both in expressivity and efficiency. It entered for the second time the international competition of SMT solvers, SMT-COMP 2010, a joint event with the SMT workshop 2010 and the FLoC federation of conferences. It performed decently against the other participating SMT solvers, and was second in several categories.
- TLAPS, the TLA$^+$ proof system (see section 5.2), was publically released for the first time in 2010. It integrates different proof backends that can be used with a declarative and hierarchical proof language and will be our main integration platform for the years to come. The first release in April consisted of the command-line tool including the back-end provers, an interpreter of the TLA$^+$ proof language, and a simple Emacs mode. The second release in October added an integration with the TLA$^+$ toolbox, based on Eclipse.

3. Scientific Foundations

3.1. Automated and interactive theorem proving

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

The VeriDis members from Saarbrücken are developing Spass [5], one of the leading automated theorem provers for first-order logic based on the superposition calculus [26]. Recent extensions to the system include the integration of dedicated reasoning procedures for specific theories, such as linear arithmetic [39], [25], that arise in verification problems resulting from concurrent and distributed algorithms. The group also studies general frameworks for the combination of theories such as the locality principle [41] and automated reasoning mechanisms these induce.

A complementary approach is being pursued in Nancy with the development of veriT [1], an SMT (satisfiability modulo theories [28]) solver that combines decision procedures for different fragments of first-order logic and that integrates an automatic theorem prover for full first-order logic. In contrast to many other SMT solvers, the veriT solver is designed to produce detailed proofs; this makes it particularly suitable as a component of a robust cooperation of deduction tools. SMT solvers can be typically used for checking finite instances of problems; first-order provers are slower but much more effective for reasoning universally over all instances of a problem.

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 the TLA$^+$ [33] prover project at the joint INRIA-MSR laboratory in Saclay whose goal is to provide an infrastructure for developing such proofs stated in a declarative proof language, and based on automatic backends.

3.2. Methodology of proved system development

Powerful theorem provers are not a panacea for system verification: their use needs to be based on a sound methodology for modeling and verifying systems. In this respect, members of VeriDis have gained expertise and recognition in developing and applying formal methods for concurrent and distributed algorithms and systems [2], [4], and we will continue to contribute to their development. In particular, the concept of *refinement* [24], [27], [37] in state-based modeling formalisms is central to our approach. Its basic idea is to derive an algorithm or implementation by providing a series of models, starting from a high-level description that precisely states the problem, and gradually adding details in intermediate models. An important goal in designing such methods is to reduce the number of generated proof obligations and/or to make them easier to establish by automatic tools. This requires taking into account specific characteristics of certain classes of systems, 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 (e.g., veriT). Eventually, properties are proved by a combination of interactive and automatic theorem proving tools (e.g., SPASS(LA)), potentially again with support of SMT procedures for specific sub-problems, or with the help of interactive proof guidance.

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 current system paradigms, including ad-hoc and overlay networks or peer-to-peer systems.

4. Application Domains

4.1. Application Domains

Our work focuses on distributed algorithms and protocols. These are or will be found at all levels of computing infrastructure, from many-core processors and systems-on-chip to wide-area networks. We are particularly interested in novel paradigms, for example ad-hoc networks that underly mobile and low-power computing or overlay networks and peer-to-peer networking that provide services for telecommunication or cloud computing services. Distributed protocols underly computing infrastructure that must be highly available and mostly invisible to the end user, therefore correctness is important. One should note that standard problems of distributed computing such as consensus, group membership or leader election have to be reformulated for the dynamic context of these modern systems. Although we are not ourselves experts in distributed algorithms, we 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 to assert under which assumptions an algorithm or system functions as required. For example, the highest levels of the Common Criteria for Information Technology Security Evaluation require code analysis, based on mathematically precise foundations. While initially the requirements of certified development have mostly been restricted to safety-critical systems, they are becoming more and more common due to the cost associated with malfunctioning system components and software.

5. Software

5.1. The veriT solver

Participants: Thomas Bouton, Diego Caminha, David Déharbe, Pascal Fontaine [correspondant], Bruno Woltzenlogel Paleo.

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 on integers and reals. It features a very efficient decision procedure for difference logic, as well as a simplex-based reasoner for full linear arithmetic. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about concepts involving sets, relations, etc. The prover can produce an explicit proof trace when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, a regression platform using INRIA's grid infrastructure is used; it allows us to extensively test the solver on thousands of benchmarks in a few minutes. In 2010 this platform was stabilized, presented at the IJCAR workshop on Pragmatic Aspects of Automated Reasoning (PAAR) [10], made available as open source under the BSD license, and is downloadable from the web site http://www.veriT-solver.org.

The veriT solver itself is also available as open source under the BSD license, and distributed through the web site http://www.veriT-solver.org. It entered for the second time the international competition of SMT solvers SMT-COMP 2010, a joint event with the SMT workshop 2010 and the FLoC federation of conferences. As in the previous competition, it performed decently against the other participating SMT solvers; it was second in several categories.

Efforts in 2010 have been focused on the extension of the expressivity of the tool (with a significant improvement in the quantifier handling), on its interface (with the adoption of the SMT-LIB 2.0 standard), and on its efficiency (which was significantly improved notably thanks to a better custom underlying SAT-solver).

Future research and implementation efforts will be directed to furthermore extend the accepted language, and increase the efficiency. 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.

5.2. The TLA+ proof system

Participants: Stephan Merz, Hernán Vanzetto.

TLAPS, the TLA$^+$ proof system, is a platform for the development and mechanical verification of TLA$^+$ proofs. It is developed at the Joint MSR-INRIA Centre. The TLA$^+$ proof language is declarative and based on standard mathematical logic; it supports hierarchical and non-linear proof construction and verification. TLAPS consists of a *proof manager* that interprets the proof language and generates a collection of proof obligations that are sent to *backend verifiers* that include theorem provers, proof assistants, SMT solvers, and decision procedures.

The first public release of TLAPS is available from http://msr-inria.inria.fr/~doligez/tlaps/, it is distributed under a BSD-like license. It handles almost all the non-temporal part of TLA$^+$ and can be used to prove safety, but not liveness properties. Currently, its backends include a tableau prover for first-order logic, an encoding of TLA$^+$ in the proof assistant Isabelle, as well as an SMT translation and a custom decision procedure for Presburger arithmetic. It is integrated with the TLA$^+$ toolbox, based on Eclipse. The main contribution of our team has been the encoding of TLA$^+$ in Isabelle, used as an automatic backend, but also to certify proofs generated by other backends.

TLAPS has been presented at IJCAR 2010 [12] and in an invited talk at ICTAC 2010 [11]. A tutorial on the system was organized as a satellite event of IFM 2010.

6. New Results

6.1. Combination of theories through the exchange of (model-)equalities

Participants: Diego Caminha, Pascal Fontaine.

The sound and complete combination of decision procedures for disjoint quantifier-free first-order languages requires the decision procedures to be able to produce entailed disjunctions of equalities: given a set of literals, the decision procedures should not only state whether the set is satisfiable or not, but also – in case the set is satisfiable – if there are disjunctions of equalities between terms that are deducible from the set. Finding such entailed disjunctions of equalities may be expensive, their number may be huge and handling all of them may become impractical. Furthermore, inspecting disjunctions of several equalities requires some case splitting on equalities in the disjunction; this may also be a factor of inefficiency.

We observed that the cooperation of decision procedures could not only be done through the exchange of deduced disjunctions of equalities, but also with the exchange of model-equalities that are guessed (with possible backtracking) by inspecting tentative models. Using these guessed equalities the expensive deduction and treatment of disjunctions of equalities is not required anymore. The case splitting is done at the SAT solver level, and only equalities are exchanged between decision procedures. These results were first presented at SBMF 2008. A more in depth presentation has been published in [6].

6.2. Compression of SMT proofs

Participants: Pascal Fontaine, Stephan Merz, Bruno Woltzenlogel Paleo.

Integrating an SMT solver in a certified environment such as an LF-style proof assistant requires the solver to output proofs. Unfortunately, those proofs may be quite large, and the overhead of rechecking the proof may account for a significant fraction of the proof time. In a paper published at SMT 2010 [19], we explore techniques for reducing the sizes of propositional proofs, which are at the core of SMT proofs. Our techniques are justified in an algebra of resolution and rely on the representation of proofs as a graph that allows us to detect the potential for reordering and combining resolution inferences.

Current and future work includes combining those techniques into a practical algorithm, implementing this algorithm, and evaluating the improvement of proof sizes.

In addition to propositional resolution proofs stemming from SMT solvers such as veriT, we have also investigated the compression of proofs in other logical calculi. In [16] we defined the CIRes method for introducing atomic cuts and demonstrated that it is capable of obtaining an exponential compression in the size and length of propositional sequent calculus proofs in the best cases. This is the theoretical maximum compression achievable by introducing only atomic cuts. Furthermore, in [22] we showed that the method also works for first-order sequent calculus proofs, being capable of compressing an exponentially growing sequence of proofs encoding the backward-chaining computation of Fibonacci numbers by a logic program into a linearly growing sequence of proofs encoding the forward-chaining computation of Fibonacci numbers.

CIRes is in fact based on CERes [35]: a method of cut-elimination that has already been implemented and used for the analysis of mathematical proofs. In [13], we describe the current status of the CERes system, focusing on recent improvements of the extraction of summarized and compressed information (in the form of Herbrand sequents) from formal proofs.

Besides our technical work on the compression of proofs, we have also explored its philosophical implications in two different areas. In [21] we have advocated for a more prominent role for (formal) proofs in the philosophy of science and in particular in the formalization of physics. In [20] we have cooperated with Ekaterina Lebedeva of the Calligramme team, in order to propose a proof-theoretical definition for the concept of implicature in the area of natural language pragmatics.

6.3. The TLA+ proof system and proof language

Participants: Stephan Merz, Hernán Vanzetto.

The TLA$^+$ proof system (see 5.2) is developed within a project at the MSR-INRIA Joint Centre in which we participate. Besides contributing to the two public releases of the system in 2010, our work on the proof system has been centered around three topics. First, the current system only supports non-temporal proofs, and we have worked on the extension of the proof language within TLA$^+$ for temporal and modal reasoning. Our objective is to obtain a sound proof calculus that is as close as possible to natural deduction, which underlies the current proof language. Second, we have prepared a significant extension of the current translation of proof obligations to back-end SMT solvers. The main difficulty has been to define and implement sort inference for (untyped) TLA$^+$ proof obligations in order to ensure the soundness of the translation, but also to improve its effectiveness (for example, by distinguishing equality over set, function, and base types). Finally, we have continued to improve the support of Isabelle/TLA$^+$ for TLA$^+$ constructs, and to augment the automatic reasoning methods within this backend. TLAPS has been presented in a conference paper and an invited talk [11], [12], and the ongoing work will be publically released in future versions of the system.

6.4. Model checking within SimGrid

Participants: Stephan Merz, Martin Quinson [of project team AlGorille], Cristián Rosa.

For several years we have cooperated with Martin Quinson from the AlGorille project team on adding model checking capabilities to the simulation platform SimGrid for message-passing distributed C programs. The expected benefit of such an integration is that programmers can complement simulation runs by exhaustive state space exploration in order to detect errors such as race conditions that would be hard to reproduce by testing. Indeed, a simulation platform provides a controlled execution environment that mediates interactions between processes, and between processes and the environment, and thus provides the basic functionality for implementing a model checker. The principal challenge is the state explosion problem, as a naive approach to systematic generation of all possible process interleavings would be infeasible beyond the most trivial programs. Moreover, it is impractical to store the set of global system states that have already been visited: the programs under analysis are arbitrary C programs with full access to the heap, making even a hashed representation of system states very difficult and costly to implement.

In 2010, we made major advances on both the theoretical and the practical side of designing and implementing a model checker for SimGrid. Given that processes interact through explicit message passing, which is ultimately implemented in the SIMIX layer of SimGrid, the model checker needs only control the possible interleavings of the operations provided at this layer (`Send`, `Recv`, `Test`, and `WaitAny`). Redundant interleavings can be ruled out by relying on Dynamic Partial-Order Reduction (DPOR) [32], which requires determining which interleaving orders may potentially lead to different results. Because we have only four primitive operations to consider, we could formally specify them in TLA$^+$ and prove independence results based on this model. This compares very favorably to a similar analysis carried out by Li et al. [36] at the MPI level, requiring more than 100 pages of TLA$^+$ specifications alone, for comparable reductions. This result has been published at a workshop [15].

In fact, the same techniques are also useful for avoiding redundant computations when performing large numbers of simulation runs. A stateless model checker relying on DPOR has now been implemented within the SimGrid platform, and a submission to a major conference is in preparation.

6.5. A new version of PlusCal

Participants: Sabina Akhtar, Stephan Merz, Martin Quinson [of project team AlGorille].

In cooperation with Martin Quinson of the AlGorille team of INRIA Nancy we have defined and implemented a high-level language for the description of concurrent and distributed algorithms. Our objective is to make formal verification techniques such as model checking available for algorithm designers without requiring them to specify algorithms in a formal modeling language such as TLA$^+$. Our work is inspired by Lamport's PlusCal [34]. Our version removes several restrictions of the original language and introduces features useful for describing and verifying models of distributed algorithms. Most importantly, processes can be nested and variables are properly scoped; this is useful for modeling concurrent execution at different levels of a hierarchy (such as threads versus processes) and for exploiting the locality of actions in order to reduce the number of interleavings that must be explored in model checking.

The design of the language and its compiler to TLA$^+$ are now stable, and two conference papers have been published [17], [8]. Most of the effort in 2010 has gone into making partial-order reduction techniques available for the verification of PlusCal algorithms. The basic idea is to adapt known techniques for static partial-order reduction; the principal difficulties are the coarser granularity and non-determinism of actions in TLA$^+$ specifications, and the absence of the notion of process. These problems are overcome by relying on static analysis of the underlying PlusCal model, whose results are passed on to the TLA$^+$ model checker.

6.6. Verification of distributed algorithms in the Heard-Of model

Participants: Henri Debrat, Stephan Merz.

Distributed algorithms are often quite subtle, both in the way they operate and in the assumptions under which they work correctly. Formal models are important for unambiguously understanding the hypotheses required, and the properties guaranteed, by a distributed algorithm. We have continued to study round-based algorithms for fault-tolerant distributed systems expressed in the Heard-Of model of Charron-Bost and Schiper [30], for which we had already proved a reduction theorem in previous work.

In 2010, our work has centered around an extension of the model and the verification framework for accomodating transmission errors (also known as Byzantine faults). We have in particular studied the well-known EIG algorithm, reformulated in the Heard-Of model. Preliminary results have been presented at a workshop [18] and in two invited talks at ICTAC 2010 and MPC/AMAST 2010. We have also continued towards a formalization of the proof of our reduction theorem in the interactive proof assistant Isabelle/HOL, which will allow us to close the gap between an operational description of the algorithm and its coarse-grained representation used for verification.

6.7. Modeling and verifying the Pastry routing protocol

Participants: Tianxiang Lu, Stephan Merz, Christoph Weidenbach.

As a significant case study for the techniques that we are developing within VeriDis, we are modeling and verifying the routing protocol of the Pastry algorithm [29] for maintaining a distributed hash table in a peer-to-peer network. As part of his PhD work (under the joint supervision of Stephan Merz and Christoph Weidenbach from MPI-INF Saarbrücken), Tianxiang Lu has developed a sizable model in TLA$^+$ of the routing protocol. The model has been validated using model checking, and several properties have been verified for small instances of the system. This work has allowed us to uncover several issues that are unclear in the existing presentations of the protocol. The model has been refined after discussions with some of the designers of the protocol and consultation of the open-source implementation FreePastry.

Tianxiang Lu has subsequently begun writing formal proofs of selected correctness properties of the protocol using TLAPS. Beyond the primary purpose of uncovering potential remaining problems with the model (or possibly the algorithm itself), these proofs also serve as a sizable case study for our tools and point to the potential for using automated provers, including veriT and Spass. Part of the work has been presented at a workshop [14].

6.8. Incremental development of distributed algorithms

Participant: Dominique Méry.

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 works help to formalize pre-existing algorithms as well as to develop new algorithms, as well as developing models for distributed systems.

Our research, carried out with Nazim Benaïssa of the MOSEL team of LORIA, is supported by the ANR project RIMEL (see 7.3 and we obtained many interesting results. More precisely, Nazim Benaïssa and Dominique Méry consider the refinement-based process for the development of security protocols. Our approach is based on the Event-B refinement, which makes proofs easier and which makes the design process faithful to the structure of the protocol as *the designer thinks of it*. We introduce the notion of mechanism related to a given security property; a mechanism can be combined with another mechanism through the double refinement process ensuring the preservation of previous security properties of mechanisms. Mechanisms and combination of mechanisms are based on Event-B models related to the security property of the current mechanism. Analyzing cryptographic protocols requires a precise model of the attacker's knowledge, within the Dolev-Yao model.

Publications on this work include project deliverables, the PhD thesis of Nazim Benaïssa, and a conference publication [9].

6.9. Formalization of automata-theoretic concepts

Participant: Stephan Merz.

In cooperation with Laurent Doyen of LSV (CNRS, ENS Cachan) and Bernard Boigelot (Liège University) we have started to work on a formalization of elementary concepts of automata theory in the interactive proof assistant Isabelle/HOL [38]. The objective of this line of research is to obtain a certified library of automata-theoretic constructions that underly decision procedures for model checking or arithmetic, building on earlier work on automata translations for temporal logic formulas [40]. In a student project with Anas Nachid from Ecole des Mines in Nancy we have in particular focused on the recent antichain techniques of Doyen et al. [31] and have formalized the underlying algebraic properties of antichains in Isabelle.

7. Contracts and Grants with Industry

7.1. Tools and Methodologies for Formal Specifications and for Proofs

Participants: Stephan Merz, Hernán Vanzetto.

We participate in the project on Tools and Methodologies for Formal Specifications and for Proofs at the MSR-INRIA Joint Centre. The objective of the project is to develop a proof environment for verifying distributed algorithms in TLA$^+$ (see also sections 5.2 and 6.3). The project in particular funds the PhD thesis of Hernán Vanzetto, which started in October 2010.

7.2. ANR project Decert

Participants: Thomas Bouton, Diego Caminha, Pascal Fontaine, Stephan Merz, Bruno Woltzenlogel Paleo.

The Decert (Deduction and Certification) project is funded by ANR from 2009–2011 within its "Domaines émergents" program. It is coordinated by the INRIA Celtique project team in Rennes, the other partners are academic teams from Orsay (INRIA Proval) and Sophia (INRIA Marelle) as well as the CEA and the Systerel company. In Nancy, the project also involves members of the Cassis team, in particular Alain Giorgetti and Christophe Ringeissen.

The objective of the project is to elaborate concepts of certification for decision procedures, including the design of appropriate certificates, the development of new certifying decision procedures, their combination, their integration with skeptical proof assistants such as Coq or Isabelle, and their use in application domains such as software verification or static analysis. The main lines of research concern questions of expressiveness vs. efficiency, certificates vs. proof objects, and the integration of certificates into verification environments. Our work within the project is related to veriT (see section 5.1), its proof production, and its integration with verification environments such as Isabelle or the TLA$^+$ proof environments (see section 5.2).

7.3. ANR project RIMEL

Participant: Dominique Méry.

The project RIMEL, carried out in cooperation with the teams led by Mohamed Mosbah and by Yves Métivier at LABRI Bordeaux and with ClearSy Systems Engineering, was extended to July 1, 2010. In Nancy, Nazim Benaïssa, Dominique Cansell, Joris Rehm, and Neeraj Singh (members of the MOSEL team) participate in the project. It focuses on the *refinement* of event-based models and aims at developing new features related to refinement, such as the reusability of refinement-based development, the composition and decomposition of models with respect to the refinement, the definition of proof-based design patterns, the integration of time constraints and probabilistic aspects, and the development of case studies, especially related to distributed systems. Probabilistic and/or timing aspects are of central importance for many distributed algorithms (such as the IEEE 1394 Tree Identification algorithm), and should therefore be integrated into the framework based on refinement. In this project, we are focusing on distributed algorithms and applications able to recover from a bad state, so-called self-healing systems. We are also interested in applications to system engineering. Our work is decomposed into several research directions:

1. Theory of refinement: integration of fairness constraints and liveness properties, probabilistic refinement and extensions of refinement scope;
2. Proof-based design patterns: a system engineering approach to justifying claims for security and trustworthiness;
3. Self-Healing Systems and Distributed Algorithms;
4. Tools and Dissemination.

8. Other Grants and Activities

8.1. European Initiatives

8.1.1. Cooperation with NUI Maynooth, Ireland

Participant: Dominique Méry.

We are involved in a bilateral research project with the National University of Ireland at Maynooth, funded within the Ulysses program between France and Ireland. The project addresses the question of formally verifying safety critical properties of already implemented software control systems, guaranteeing their reliability and safety. In particular, we address the following questions: What is the best methodology for generating a formal system requirements document (written in Event-B) for an already existing tram control system? What is the relationship between Event-B and Programmable Logic? How effectively can we support the formal translation of a system specification written in Event-B to its implementation written in programmable logic? Can we demonstrate that this formal transformation preserves the safety critical properties as specified for an existing tram control system? A combination of reverse engineering and refinement techniques will be used to prove the safety critical properties of a tram control system, generating a suite of proof based patterns that may be used in the verification of safety critical properties of similar systems. Case studies involving subsystems of the tram control system will be used to develop Master level courses, ensuring technology transfer between industry and the classroom, and vice versa. A visit of Dominique Méry in February led to a series of lectures in the master program; Dominique Méry is completing models for ensuring the quality of produced codes. During a reciprocal visit of Rosemary Monahan of NUI Maynooth in October, she gave a tutorial on the verification of C# programs using Spec# and Boogie 2.

8.1.2. Formalizing automata theory

Participants: Pascal Fontaine, Stephan Merz.

We have obtained a starting grant within the COLOR program of INRIA Nancy to work with Bernard Boigelot and Pierre Wolper of Liège University on the formalization of certain automata-theoretic constructions in the interactive proof assistant Isabelle/HOL. The objective is to obtain reference implementations of basic automata algorithms used in verification whose correctness is proved in Isabelle, and for which code is automatically generated. The project concerns, on one hand, the study of antichain techniques for finite automata (see section 6.9). On the other hand, we intend to study real-vector automata used for decision procedures for integer and real arithmetic.

The work on formalizing automata constructions in Isabelle is also part of an informal cooperation with the groups of Jan-Georg Smaus in Freiburg and Tobias Nipkow and Javier Esparza at TU Munich. Our German partners have obtained a multi-site national research (DFG) grant on this topic, and Stephan Merz is an associated researcher of the consortium.

8.2. International Initiatives

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

Participants: Diego Caminha, David Déharbe, Pascal Fontaine, Stephan Merz, Bruno Woltzenlogel Paleo.

The team has a close working relationship with a team at Universidade Federal do Rio Grande de Norte (UFRN), Brazil, and more particularly with Prof. Anamaria Martins Moreira and Prof. David Déharbe. Four exchanges occurred in 2010. Stephan Merz visited UFRN for 10 days from August 25 to September 4, Pascal Fontaine for two weeks from October 18 to 30, and Sabina Akhtar from November 6 to 12. David Déharbe was in Nancy for 2 weeks, from November 25 to December 9. The project is centered around the development and applications of the veriT solver (section 5.1), of which David Déharbe and Pascal Fontaine are the main developers. Diego Caminha was previously a student at UFRN and joined the VeriDis team as a PhD student (defense planned for early 2011). Our cooperation is supported by the INRIA-CNPq project SMT-SAVeS from 2010 throughout 2012.

9. Dissemination

9.1. Animation of the scientific community

- Pascal Fontaine served on the program committees of ICTAC 2010 and PAAR 2010.
- Dominique Méry served on the program committees of ICFEM 2010, CAL 2010, and IFM 2010. He has organized several events related to formal methods: The B Method – from Research to Teaching on June 15, 2010 in Nantes, France, a special session in the ISOLA conference and IFM 2010 jointly with Stephan Merz.
- Stephan Merz served on the program committees of AVOCS 2010, SBMF 2010, the Grande Région Security Day 2010 in Saarbrücken, and co-chaired the program committee of IFM 2010 together with Dominique Méry. He was an invited speaker at the Amir Pnueli Memorial Days (New York), at AMAST/MPC 2010 (Québec), and at ICTAC 2010 (Natal, main conference and summer school). He organized IFM 2010 jointly with Dominique Méry and the VTSA 2010 Summer School on Verification Technology, Systems, and Applications together with Christoph Weidenbach of MPI-INF Saarbrücken and Jun Pang of the University of Luxemburg.

9.2. Theses, habilitations, academic duties

- Pascal Fontaine is a member of an international working group designing the proof format for SMT solvers. He is the head of an undergraduate program (Licence Miage) at Nancy 2 University.
- Dominique Méry is
  - a member of the IFIP Working Group 1.3 on *Foundations of System Specification*,
  - the Head of the Doctoral School IAEM Lorraine for the four universities of Lorraine,
  - a member of the scientific council of the LORIA laboratory,
  - an expert for the French Ministry of Education (DS9),
  - an expert for the French Agence Nationale de la Recherche (ANR) and AERES.
  - the director of international affairs at ESIAL Nancy, and
  - the president of the APCB association.
- Stephan Merz is
  - a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*,
  - the delegate for international relations at LORIA and INRIA Nancy,
  - an elected member of the evaluation committee of INRIA, he participated in the hiring committee for junior researchers at INRIA Saclay,
  - a nominated member of the Section 7 of the Comité National de la Recherche Scientifique,
  - the INRIA representative in the Scientific Directorate of the International Computer Science Meeting Center in Dagstuhl,
  - an expert for the French Agence Nationale de la Recherche (ANR), AERES, and the German DAAD, and
  - served as external evaluator for a PhD committee at LMU Munich.

9.3. Teaching

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

- Dominique Méry gave courses in the Master's program in Nancy on: formal system engineering, modelling and verification of systems, theoretical computer science, development of software systems, distributed algorithms.
- Stephan Merz, together with Laurent Vigneron, gave a course on algorithmic verification in the Master's program in Nancy.

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, p. 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, Berlin-Heidelberg, 2008, p. 47–152.

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

[4] 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, Berlin-Heidelberg, 2008, p. 401–451.

[5] C. WEIDENBACH, D. DIMOVA, A. FIETZKE, M. SUDA, P. WISCHNEWSKI. *SPASS Version 3.5*, in "22nd Intl. Conf. Automated Deduction (CADE-22)", Montreal, Canada, R. SCHMIDT (editor), Lecture Note in Computer Science, Springer, 2009, vol. 5663, p. 140-145.

Publications of the year

Articles in International Peer-Reviewed Journal

[6] D. CAMINHA B. DE OLIVEIRA, D. DÉHARBE, P. FONTAINE. *Combining decision procedures by (model-)equality propagation*, in "Science of Computer Programming", 2010, http://hal.inria.fr/inria-00543801/en.

[7] H. ZHANG, S. MERZ, M. GU. *Specifying and Verifying PLC systems with TLA+: a case study*, in "Computers & Mathematics with Applications", August 2010, vol. 60, n^o 3, p. 695-705 [*DOI :* 10.1016/J.CAMWA.2010.05.017], http://hal.inria.fr/hal-00516785/en.

International Peer-Reviewed Conference/Proceedings

[8] S. AKHTAR, S. MERZ, M. QUINSON. *A High-Level Language for Modeling Algorithms and their Properties*, in "13th Brazilian Symposium on Formal Methods - SBMF'2010", Brazil Natal, November 2010, http://hal.inria.fr/inria-00537779/en.

[9] N. BENAÏISSA, D. MÉRY. *Proof-Based Design of Security Protocols*, in "5th International Computer Science Symposium in Russia, CSR 2010", Russian Federation Kazan, E. W. MAYR (editor), Lecture Notes in Computer Science, Springer, June 2010, vol. 6072, p. 25-36, http://hal.inria.fr/inria-00542919/en.

[10] T. BOUTON, D. CAMINHA B. DE OLIVEIRA, D. DÉHARBE, P. FONTAINE. *GridTPT: a distributed platform for Theorem Prover Testing*, in "2nd Workshop on Practical Aspects of Automated Reasoning (PAAR)", United Kingdom Edinburgh, 2010, http://hal.inria.fr/inria-00543805/en.

[11] K. CHAUDHURI, D. DOLIGEZ, L. LAMPORT, S. MERZ. *The TLA+ Proof System: Building a Heterogeneous Verification Platform*, in "International Conference on Theoretical Aspects of Computing - ICTAC 2010", Brazil Natal, A. CAVALCANTI, D. DÉHARBE, M.-C. GAUDEL, J. WOODCOCK (editors), Lecture Notes in Computer Science, Springer, 2010, vol. 6255, 44, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-14808-8_3], http://hal.inria.fr/inria-00521886/en.

[12] K. CHAUDHURI, D. DOLIGEZ, L. LAMPORT, S. MERZ. *Verifying Safety Properties With the TLA+ Proof System*, in "Fifth International Joint Conference on Automated Reasoning - IJCAR 2010", United Kingdom Edinburgh, J. GIESL, R. HÄHNLE (editors), Lecture Notes in Artificial Intelligence, Springer, 2010, vol. 6173, p. 142–148, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-14203-1_12], http://hal.inria.fr/inria-00534821/en.

[13] T. DUNCHEV, A. LEITSCH, T. LIBAL, D. WELLER, B. WOLTZENLOGEL PALEO. *System Description: The Proof Transformation System CERES*, in "International Joint Conference on Automated Reasoning", United Kingdom Edinburgh, J. GIESL, R. HÄHNLE (editors), Lecture Notes in Computer Science / Lecture Notes in ArtificiaI Intelligence, Springer, 2010, vol. 6173, p. 427-433, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-14203-1_36], http://hal.inria.fr/hal-00545482/en.

[14] T. LU, S. MERZ, C. WEIDENBACH. *Model Checking the Pastry Routing Protocol*, in "10th International Workshop Automated Verification of Critical Systems", Germany Düsseldorf, J. BENDISPOSTO, M. LEUSCHEL, M. ROGGENBACH (editors), Universität Düsseldorf, September 2010, p. 19-21, short communication, http://hal.inria.fr/inria-00540811/en.

[15] C. ROSA, S. MERZ, M. QUINSON. *A Simple Model of Communication APIs – Application to Dynamic Partial-order Reduction*, in "10th International Workshop on Automated Verification of Critical Systems - AVOCS 2010", Germany Düsseldorf, September 2010, http://hal.inria.fr/inria-00532889/en.

[16] B. WOLTZENLOGEL PALEO. *Atomic Cut Introduction by Resolution: Proof Structuring and Compression*, in "16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning - LPAR-16", Senegal Dakar, Lecture Notes in Computer Science, Springer, April 2010, vol. 6355, p. 463-480, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-17511-4_26], http://hal.archives-ouvertes.fr/hal-00545473/en/.

National Peer-Reviewed Conference/Proceedings

[17] S. AKHTAR, S. MERZ, M. QUINSON. *Extending PlusCal: A Language for Describing Concurrent and Distributed Algorithms*, in "Actes des deuxièmes journées nationales du Groupement De Recherche CNRS du Génie de la Programmation et du Logiciel", France Pau, E. CARIOU, L. DUCHIEN, Y. LEDRU (editors), March 2010, http://hal.inria.fr/inria-00544137/en.

Workshops without Proceedings

[18] H. DEBRAT, B. CHARRON-BOST, S. MERZ. *Formal Verification of Consensus Algorithms in a Proof Assistant*, in "2010 Grande Region Security and Reliability Day", Germany Saarbrücken, M. BACKES, R. KÜSTERS (editors), March 2010, http://hal.inria.fr/inria-00539899/en.

[19] P. FONTAINE, S. MERZ, B. WOLTZENLOGEL PALEO. *Exploring and Exploiting Algebraic and Graphical Properties of Resolution*, in "8th International Workshop on Satisfiability Modulo Theories - SMT 2010", United Kingdom Edinburgh, July 2010, http://hal.inria.fr/inria-00544658/en.

[20] B. WOLTZENLOGEL PALEO, E. LEBEDEVA. *Using Proofs to Compute Implicatures [Abstract]*, in "Computability in Europe", Portugal Ponta Delgada, June 2010, http://hal.inria.fr/hal-00545494/en.

[21] B. WOLTZENLOGEL PALEO. *Physics and Proof Theory*, in "International Workshop on Physics and Computation", Egypt Luxor, August 2010, http://hal.inria.fr/hal-00545462/en.

[22] B. WOLTZENLOGEL PALEO. *Proof Compression with the CIRes Method [Abstract]*, in "Computability in Europa", Portugal Ponta Delgada, June 2010, http://hal.inria.fr/hal-00545496/en.

Books or Proceedings Editing

[23] D. MÉRY, S. MERZ (editors). *Integrated Formal Methods*, Lecture Notes in Computer Science, Springer, October 2010, vol. 6396 [*DOI :* 10.1007/978-3-642-16265-7], http://hal.inria.fr/inria-00539785/en.

References in notes

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

[25] 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, p. 84-99.

[26] 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, p. 217–247.

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

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

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

[30] B. CHARRON-BOST, A. SCHIPER. *The Heard-Of model: computing in distributed systems with benign faults*, in "Distributed Computing", 2009, vol. 22, n^{o} 1, p. 49-71.

[31] L. DOYEN, J.-F. RASKIN. *Antichain Algorithms for Finite Automata*, in "16th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010)", Paphos, Cyprus, J. ESPARZA, R. MAJUMDAR (editors), Lecture Notes in Computer Science, Springer, 2010, vol. 6015, p. 2-22.

[32] C. FLANAGAN, P. GODEFROID. *Dynamic partial-order reduction for model checking software*, in "32nd ACM Symp. Principles of Programming Languages (POPL 2005)", Long Beach, CA, U.S.A., J. PALSBERG, M. ABADI (editors), ACM, 2005, p. 110–121.

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

[34] L. LAMPORT. *Checking a Multithreaded Algorithm with +CAL*, in "20th Intl. Symp. Distributed Computing (DISC 2006)", Stockholm, Sweden, S. DOLEV (editor), Lecture Notes in Computer Science, 2006, vol. 4167, p. 151–163.

[35] A. LEITSCH, M. BAAZ. *Methods of Cut-Elimination*, Springer, 2011, To appear.

[36] G. LI, R. PALMER, M. DELISI, G. GOPALAKRISHNAN, R. M. KIRBY. *Formal specification of MPI 2.0: Case study in specifying a practical concurrent programming API*, in "Sci. Comput. Program.", 2011, vol. 76, n^{O} 2, p. 65-81.

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

[38] T. NIPKOW, L. PAULSON, M. WENZEL. *Isabelle/HOL. A Proof Assistant for Higher-Order Logic*, Lecture Notes in Computer Science, Springer Verlag, 2002, n^{O} 2283.

[39] 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, p. 18-33.

[40] A. SCHIMPF, S. MERZ, J.-G. SMAUS. *Construction of Büchi Automata for LTL Model Checking Verified in Isabelle/HOL*, in "22nd Intl. Conf. Theorem Proving in Higher-Order Logics (TPHOLs 2009)", Munich, Germany, T. NIPKOW, C. URBAN (editors), Lecture Notes in Computer Science, Springer, 2009, vol. 5674, p. 424-439.

[41] 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, p. 47-71, Invited paper.