Activity Report 2011

Team VERIDIS

Verification of Distributed Systems

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

RESEARCH CENTER
Nancy - Grand Est

THEME
Programs, Verification and Proofs

Table of contents

Team VERIDIS

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

The VeriDis team is a joint proposal between members of the Mosel team at LORIA, Nancy, France, and members of the Automation of Reasoning group at Max-Planck Institute for Informatics in Saarbrücken, Germany. The proposal was evaluated positively in spring 2011 by the group of experts named by INRIA, but has not yet been created as a joint team. Consequently, this report only presents work involving members of the team in Nancy.

1. Members

Research Scientist

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

Faculty Members

Marie Duflot-Kremer [Associate Professor, Université Henri Poincaré Nancy 1, since September 2011]
Pascal Fontaine [Associate Professor, Université Nancy 2, on leave at INRIA since 09/2011]
Dominique Méry [Professor, Université Henri Poincaré Nancy 1, HdR]

External Collaborator

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

PhD Students

Sabina Akhtar [Université Henri Poincaré Nancy 1, since 09/2008, grant from foreign government]
Manamiary Andriamiarina [Université Henri Poincaré Nancy 1, since 10/2010]
Diego Caminha Barbosa de Oliveira [Université Nancy 2, until 03/2011]
Henri Debrat [Université Henri Poincaré Nancy 1, since 10/2009, joint supervision with Bernadette Charron-Bost]
Tianxiang Lu [Universität des Saarlands, since 05/2009, joint supervision with Christoph Weidenbach]
Cristián Rosa [Université Henri Poincaré Nancy 1, until 10/2011, joint supervision with Martin Quinson of AlGorille team]
Hernán-Pablo Vanzetto [Université Henri Poincaré Nancy 1, since 10/2010, joint supervision with Kaustuv Chaudhuri]

Post-Doctoral Fellow

Bruno Woltzenlogel Paleo [until 03/2011]

Administrative Assistant

Isabelle Herlich [shared with teams AlGorille, Alice, Score]

Others

Hernán Ponce de Leon [Universidad Nacional de Rosario, Argentina, student intern 04-07/2011]
Julien Perugini [ESIAL, student intern 06-08/2011]
Pierre Savonitto [ESIAL, student intern 06-08/2011]

2. Overall Objectives

2.1. Introduction

VeriDis was created in January 2010 as a local team of INRIA Nancy Grand-Est. The scientific proposal includes members of the MOSEL group of LORIA, the computer science laboratory in Nancy and members of the Automation of Logic Research Group at Max-Planck Institut for Informatics in Saarbrücken, led by Christoph Weidenbach. This joint proposal was positively evaluated by the scientific experts nominated by INRIA, and the *comité des projets* of INRIA Nancy recommended in June 2011 that the team be created.

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. Ideally, systems and their properties would be specified in high-level, expressive languages, errors in specifications would be discovered automatically, and finally, full verification could also be performed completely automatically. Due to the inherent complexity of the problem this cannot be achieved in general. However, we have observed important advances in automated and interactive theorem proving in recent years. We are particularly interested in the integration of different deduction techniques and tools, including the combination of relevant theories such as arithmetic in automated theorem proving. These advances suggest that a substantially higher degree of automation can be achieved in system verification over what is available in today's verification tools.

VeriDis proposes to exploit and further develop automation in system verification, and to apply its techniques within the context of concurrent and distributed algorithms, which are by now ubiquitous and whose verification is a big challenge. Concurrency problems are central to the development and verification of programs for multi- and many-core architectures, and distributed computation underlies the paradigms of grid and cloud computing. Typical application problems that we 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. 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

- Marie Duflot-Kremer joined VeriDis in September 2011. Previously at University Paris Est Créteil, she is an assistant professor at University Henri Poincaré Nancy 1. Her research is centered around statistical model checking and the verification of probabilistic systems.
- The veriT solver (see section 5.1) entered for the third time the international competition of SMT solvers, SMT-COMP 2011, a joint event with the SMT workshop 2011 and the CAV conference. It implemented a new original technique (presented at CADE 2011) that greatly improves efficiency on some categories of benchmarks. Several competitors also implemented this technique, as for instance the winner of the competition on those categories (Z3).
- Pascal Fontaine (VeriDis) and Aaron Stump (University of Iowa) organized the first workshop on Proof eXchange for Theorem Proving, co-located with CADE 2011. The workshop was well attended and we believe that this series of events will stimulate research in the area, and will lead to important improvement in reasoning techniques.

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 Nancy develop veriT [1], an SMT (satisfiability modulo theories [24]) solver that combines decision procedures for different fragments of first-order logic and that integrates an automatic theorem prover for full first-order logic. The veriT solver is designed to produce detailed proofs; this makes it particularly suitable as a component of a robust cooperation of deduction tools.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction. Members of VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we participate in a project at the joint INRIA-MSR laboratory in Saclay on the development of methods and tools for the formal proof of TLA$^+$ [28] specification. Our prover relies on a declarative proof language and includes several automatic backends [3].

3.2. Methodology of proved system development

Powerful theorem provers are not a panacea for system verification: 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], [5], and we will continue to contribute to their development. In particular, the concept of *refinement* [21], [23], [31] 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, 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. We are not ourselves experts in the design of distributed algorithms, but work together with domain experts on the modeling and verification of these protocols. These cooperations help us focus on concrete algorithms and ensure that our work is relevant to the distributed algorithm community.

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

5. Software

5.1. The veriT solver

Participants: Diego Caminha Barbosa de Oliveira, 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.

The veriT solver is available as open source under the BSD license, and distributed through the web site http://www.veriT-solver.org. It entered for the third time the international competition of SMT solvers SMT-COMP 2011, a joint event with the SMT workshop 2011 and the CAV conference. As in the previous competitions, it performed decently against the other participating SMT solvers. It embeds an original symmetry reduction technique that greatly improved its efficiency on some categories of formulas. This technique was immediately incorporated also in other competing solvers, in particular Z3 (Microsoft) and CVC3 (University of New-York and University of Iowa).

Efforts in 2011 have been focused on the extension of the expressiveness of the tool (with improvements in the handling of quantifiers), and on its efficiency (which was significantly improved at different levels, including a purpose-built SAT solver underlying veriT). A lot of work was also devoted to improve the proof production of the tool, with the definition of a precise proof language. This proof language has been presented to the community as a standard for describing SMT proofs [17]. We are collaborating on this with Laurent Théry and Benjamin Grégoire (Marelle, INRIA Sophia-Antipolis), Laurent Voisin (Systerel), and Frédéric Besson (Celtique, INRIA Rennes).

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.

The software will be supported by an INRIA ADT, which will start at the beginning of 2012.

5.2. The TLA+ proof system

Participants: Stephan Merz, Hernán-Pablo Vanzetto.

TLAPS, the TLA$^+$ proof system, is a platform for developing and mechanically verifying 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.

TLAPS is publically available at http://msr-inria.inria.fr/~doligez/tlaps/, it is distributed under a BSD-like license. It handles the non-temporal part of TLA$^+$ with the exception of computing enabledness predicates and can currently be used to prove safety, but not liveness properties. Its backends include a tableau prover for first-order logic, an encoding of TLA$^+$ in the proof assistant Isabelle, as well as an SMT translation and a custom decision procedure for Presburger arithmetic. Our main contribution in 2011 has been the implementation of a new SMT backend that handles formulas including linear arithmetic, elementary set theory, functions, tuples, and records (see section 6.4). Other efforts in 2011 concerned improvements and stabilization of the fingerprinting technique that avoids reproving proof obligations that have remained unchanged since a previous prover run.

6. New Results

6.1. Using symmetries in SMT

Participants: David Déharbe, Pascal Fontaine, Bruno Woltzenlogel Paleo.

Methods exploiting problem symmetries have been very successful in several areas including constraint programming and SAT solving. We propose a similar technique for enhancing the performance of SMT-solvers by detecting symmetries in the input formulas and using them to prune the search space of the SMT algorithm. This technique is based on the concept of (syntactic) invariance by permutation of constants. An algorithm for solving SMT by taking advantage of such symmetries is presented. The implementation of this algorithm in the SMT-solver veriT results in an impressive improvement of veriT's performances on the SMT-LIB benchmarks that places it ahead of the winners of the last editions of the SMT-COMP contest in the QF_UF category.

This technique has immediately been adopted by the SMT community. For instance, we are aware that Z3 (Microsoft) and CVC3 (University of New-York and University of Iowa) implemented this technique for the 2011 competition.

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 previous work, we proposed a technique for reducing the sizes of propositional proofs based on the analysis of resolution graphs, which were justified in an algebra of resolution. Unfortunately, the complexity of these techniques turned out to be prohibitive. In a paper published at CADE 2011 [11], we give practical algorithms for more restricted compression techniques and validate them on standard benchmarks. Our algorithms significantly improve state-of-the-art proof compression algorithms and achieve better reduction of proof sizes, often by 30%.

6.3. Combination of decision procedures

Participant: Pascal Fontaine.

We investigate the theoretical limits of combining decision procedures and reasoners, as these are important for the development of the veriT solver (see section 5.1). It has long been known that it is possible to extend any decidable language (subject to a minor requirement on cardinalities) with predicates described by a Bernays-Schönfinkel-Ramsey theory (BSR). A formula belongs to the BSR decidable fragment if it is a conjunction of universal, function-free formulas. As a consequence of this theoretical result, it is possible to extend a decidable quantifier-free language with sets and set operators, relations, orders and similar concepts. This can be used to significantly extend the expressivity of SMT solvers. In previous work, we had generalized this result to the decidable first-order class of monadic predicate logic, and to the two-variable fragment. In 2011, in cooperation with Carlos Areces from Universidad Nacional de Córdoba, Argentina, we showed that two other important decidable fragments (namely the Ackermann fragment, and several guarded fragments) are also easily combinable. This result was presented at the FroCoS Conference 2011 [8], as well as at the SMT'2011 workshop (joint with the Conference on Computer Aided Verification, CAV 2011).

6.4. Encoding TLA+ proof obligations for SMT solvers

Participants: Stephan Merz, Hernán-Pablo Vanzetto.

The TLA$^+$ proof system TLAPS (see 5.2) is being developed within a project at the MSR-INRIA Joint Centre in which we participate. The original release of TLAPS contained an SMT backend that handled quantifier-free proof obligations in linear arithmetic and that was occasionally useful, given that the other backends perform quite poorly on formulas involving arithmetic. However, TLA$^+$ proof obligations usually mix arithmetic with other theories, in particular set theory, functions, records, and tuples. We propose a new encoding of TLA$^+$ sequents in SMT-LIB, the generic input language of SMT solvers. The main challenge has been to design a sound translation from untyped TLA$^+$ to the multi-sorted first-order logic that underlies SMT-LIB. We have developed a type system and a type inference algorithm that assigns SMT-LIB sorts to symbols and terms in the input formula, based on "typing assumptions" among the hypotheses present in the proof obligation.

The translation has been validated over several existing examples, yielding significant reductions in proof sizes. For example, the new backend can automatically verify the main invariant of a parameterized version of the Bakery algorithm, which previously required a few hundred lines of interactive proof. Similarly, an existing proof about a security architecture [33] has been reduced by about 90%. The backend has been integrated in TLAPS and has been presented at a workshop [19].

6.5. 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 the 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 it difficult and costly to store global states and to determine if two states are equal.

We have implemented a stateless model checker within the SimGrid platform, for verifying safety properties of distributed C programs that communicate by message passing. The visible actions correspond to the communication events, at which points programs can be interrupted by the simulation core. In order to mitigate state explosion, the exploration relies on Dynamic Partial-Order Reduction (DPOR) that avoids exploring redundant interleavings corresponding to the same global happens-before relation. We have identified four primitive communication actions, in terms of which the different message-passing libraries provided by SimGrid can be implemented, and have proved independence theorems for these primitives that underly our DPOR exploration algorithm. We thus obtain a small kernel that supports different communication APIs; nevertheless, practical evaluations yield similar reductions as those obtained by Li et al. [30] for a much more detailed analysis of a fragment of the MPI library.

The model checker SimGridMC is now part of the SimGrid platform and allows programmers to either perform simulation or model checking runs based on the same source code. It has allowed us to discover a non-trivial bug in an implementation of the Chord algorithm for realizing a distributed hashtable over a P2P network. A conference paper has been published at FORTE 2011 [13]. Cristián Rosa successfully defended his PhD thesis [7] in October 2011, which also proposes efficient techniques for parallelizing simulation runs in SimGrid. Marion Guthmuller has explored extensions of our model checking algorithm for verifying liveness properties, and has started working on her PhD thesis in this area in the fall of 2011.

6.6. 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 work is inspired by Lamport's PlusCal [29], but extends it for the modeling and verification of distributed algorithms. In particular, 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).

In 2011, the main effort has gone into designing partial-order reduction techniques for model checking PlusCal algorithms, which exploit the locality information present in the models. In particular, we have defined predicates that ensure the independence of two (blocks of) statements and adapted the TLC model checker to implement static partial-order reduction. Sabina Akhtar prepares her PhD thesis manuscript, and the thesis defense is planned for spring 2012.

6.7. 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 required for their correctness. Formal models are important for unambiguously understanding the hypotheses and the properties of a distributed algorithm. We focus on the verification of round-based algorithms for fault-tolerant distributed systems expressed in the Heard-Of model of Charron-Bost and Schiper [26], for which we had already proved a reduction theorem in previous work.

In 2011, we have extended our previous results to the case of Byzantine errors where values may be received that do not correspond to those that should have been computed by the sender process (for example because of an intermittent fault in the sender process, a malicious process, or a value-changing error in the transmission channel). We have formalized a corresponding extension of the Heard-Of model in Isabelle/HOL, and have verified three Byzantine Consensus algorithms (EIG, ATE and UTE) within this framework. These results have been presented at SSS 2011 [9].

6.8. Modeling and verifying the Pastry routing protocol

Participants: Tianxiang Lu, Stephan Merz.

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 [25] 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 TLA$^+$ model of the Pastry routing protocol, which has uncovered several issues in the existing presentations of the protocol in the literature, and in particular a loophole in the join protocol that had been fixed by the algorithm designers in a technical report that appeared after the publication of the original protocol.

In 2011, we have worked towards a correctness proof of the routing protocol. We have in particular identified a number of candidate invariants that have been validated by extensive model checking over finite instances and for which we have formally proved that their validity would imply the correctness of the protocol. Our proofs are carried out in TLAPS (section 5.2) and represent a sizable case study for the different proof tools of the proof system. Our results have been presented at FORTE 2011 [12].

6.9. Incremental development of distributed algorithms

Participants: Dominique Méry, Manamiary Andriamiarina.

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, develop new algorithms, as well as develop models for distributed systems.

Our research, carried out with Mohammed Mosbah and Mohammed Tounsi from the LABRI laboratory, was supported by the ANR project RIMEL until 2010 and we are maintaining a joint project B2VISIDIA with LABRI on these topics. More concretely, we aim at an integration of the correct-by-construction refinement-based approach into the *local computation* programming model. The team of LABRI develops an environment called VISIDIA that provides a toolset for developing distributed algorithms expressed as a set of rewriting rules of graph structures. The simulation of rewriting rules is based on synchronization algorithms and we have developed these algorithms by refinement.

Synchronization algorithms [14] are mandatory for simulating local computation models of distributed algorithms. Therefore, correctness of these algorithms becomes crucial, because it gives confidence that local computations are simulated as designed and do not behave harmfully. However, these algorithms are often very complex to prove correct since they integrate both distributed and probabilistic aspects. We derive proofs of synchronization algorithms upon which the correct-by-construction paradigm depends; the latter is supported by a progressive and incremental process controlled by the refinement techniques. We illustrate our approach by examples such as the Handshake and the LC1 algorithms. These algorithms are designed for an asynchronous distributed network of anonymous processes that communicate by message passing.

A second contribution is related to the integration of probabilistic arguments when reasoning about the design of distributed programms. We particularly focus [20] on probabilistic aspects of distributed algorithms related to termination, e.g. the choice between two delays in the case of communication protocols like IEEE 1394 (FireWire), or the choice between several colors for vertex coloring algorithms. We have in particular applied this approach to developing probabilistic distributed graph coloring algorithms (also called vertex coloring algorithms), based on an algorithm developed by Métivier et al. [32], using the Event B and probabilistic Event B methods.

A third contribution takes into account the modification of links between nodes in a graph modelling a network. We present [15] an incremental formal development of the Dynamic Source Routing (DSR) protocol in Event-B. DSR is a reactive routing protocol, which finds a route for a destination on demand, whenever communication is needed. Route discovery is an important task of any routing algorithm and its formal specification is a challenging problem in itself. The specification is performed in a stepwise manner by introducing more advanced routing components between the abstract specification and topology. It is verified through a series of refinements. The specification includes safety properties as a set of invariants, and liveness properties that characterize when the system reaches stable states. We establish these properties by proof of invariants, event refinement and deadlock freedom. The consequence of this incremental approach helps us achieve a high degree of automatization. Our approach can be useful for formalizing and developing other kinds of reactive routing protocols such as AODV.

6.10. Bounding message length in attacks against security protocols

Participant: Marie Duflot-Kremer.

Security protocols are short programs that describe communication between two or more parties in order to achieve security goals. Despite the apparent simplicity of such protocols, their verification is a difficult problem and has been shown to be undecidable in general. This undecidability comes from the fact that the set of executions to be considered is of infinite depth (an infinite number of protocol sessions can be run) and infinitely branching (the intruder can generate an unbounded number of distinct messages). Several attempts have been made to tackle each of these sources of undecidability. Together with Myrto Arapinis, we have shown [22] that, under a syntactic and reasonable condition of "well-formedness" on the protocol, we can get rid of the infinitely branching part. More precisely we proved that as far as the secrecy property is considered and for a well-formed protocol, we just need to consider well-typed attacks, with a strong typing system. This result directly implies that the messages to be considered are of bounded length. We are currently working on a journal version of this result that extends the set of security properties to which the result is applicable, in particular including authentication properties.

6.11. Formally verified decision procedures for finite automata

Participants: Stephan Merz, Julien Perugini, Hernán Ponce de Leon, Pierre Savonitto.

Decision problems in the theory of finite automata underly verification algorithms in model checking and decision procedures for fragments of arithmetic. We are interested in developing a certified library of automata-theoretic constructions within a trusted interactive proof assistent such as Isabelle. In 2011, two student projects addressed such problems.

Julien Perugini and Pierre Savonitto formalized a decision procedure for the universality problem of finite automata based on the antichain technique suggested by Doyen et al. [27] and verified its correctness in Isabelle/HOL. They then verified a list-based implementation of that algorithm, using the Isabelle Collections Framework, which provides pre-proved data structures for generating executable implementations. Future work should address efficiency issues by adopting better suited data structures.

During his internship, Hernán Ponce de Leon formalized and verified an automaton-based decision procedure for Presburger arithmetic over the integers, based on a previous encoding of a similar procedure restricted to natural numbers.

7. Contracts and Grants with Industry

7.1. ANR project DeCert

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

The DeCert (Deduction and Certification) project is being funded by ANR from 2009–2012 within its "Domaines émergents" program. It is coordinated by the Celtique project team of INRIA Rennes, the other partners are academic teams from INRIA Saclay (Proval) and INRIA Sophia Antipolis (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 study certified 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).

7.2. Tools and Methodologies for Formal Specifications and for Proofs

Participants: Stephan Merz, Hernán-Pablo 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 6.4). The project in particular funds the PhD thesis of Hernán Vanzetto.

7.3. Diagnosis of errors in network controlled systems

Participants: Diego Caminha Barbosa de Oliveira, Pascal Fontaine, Stephan Merz.

In an exploratory project with Westinghouse France, we studied the possibility of using formal verification technology (in particular model checking and SAT/SMT solving) for diagnosing possibly transient faults in communication networks. The diagnosis is based on logs that are generated by periodic self tests. In particular, the SAT solver of veriT has been interfaced with Matlab so that it can be used by our industrial partner for determining causes of certain permanent faults. We have also used Uppaal to model a simplified version of a protocol used by our industrial partner in order to determine timing intervals for the occurrence of faults detected in logs.

8. Partnerships and Cooperations

8.1. European Initiatives

8.1.1. Cooperation with NUI Maynooth, Ireland

We are involved in a bilateral research project with the National University of Ireland at Maynooth, funded by the Ulysses program between France and Ireland. The project addresses the question of formally verifying safety critical properties of 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 are 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. Visits of Dominique Méry in February, August and December led to a series of lectures in the master program and in a Summer School organised by NUI Maynooth; 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.

8.2. International Initiatives

8.2.1. INRIA International Partners

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

VeriDis 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. Two long exchanges took place in 2011. Bruno Woltzenlogel Paleo visited UFRN for one month in March, and David Déharbe visited VeriDis from June 20 to July 20 as an INRIA invited researcher. 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 prepared his PhD thesis with the VeriDis team. Our cooperation is also supported by the INRIA-CNPq project SMT-SAVeS from 2010 throughout 2012.

8.2.1.2. Cooperation with Tiaret University

Mostapha Belardi (Université Ibn Khaldoun de Tiaret), Camel Tanougast (Univ. Paul Verlaine, Metz), Dominique Méry and Stephan Merz have started a joint project entitled *CIPRONoC : Conception Incrémentale Prouvée pour pROtotypage rapide de NoC Tolérant aux Fautes à base de technologie FPGA*. The project is sponsored by the STIC Algérie program.

8.2.2. Visits of International Scientists

8.2.2.1. INRIA Internship program

Hernán Ponce de Leon (from April 2011 until August 2011)

Subject: Formally Verified Automata Construction for Real Linear Equations

Institution: Universidad Nacional de Rosario (Argentina)

8.2.2.2. Invited scientists

David Déharbe from Universidade Federal do Rio Grande de Norte, Brazil, visited VeriDis from June 20 to July 20 as an INRIA invited researcher. The work resulted in several improvements of the veriT solver and contributed to its integration within the toolsets for the B and TLA$^+$ methods.

9. Dissemination

9.1. Animation of the scientific community

- Pascal Fontaine co-chaired the program committee of PxTP 2011 and served on the program committee of ICTAC 2011. He is a member of an international working group designing the proof format for SMT solvers.
- 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,
  - head of the Formal Methods department of the LORIA laboratory,
  - an expert for the French Ministry of Education (DS9),
  - an expert for the French Agence Nationale de la Recherche (ANR) and AERES.
  - the director of international affairs at ESIAL Nancy, and
  - the president of the APCB association.
  - He served on the program committees of ICFEM 2011 and FHIES 2011.
- The academic duties of Stephan Merz include:
  - member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*,
  - elected member of the evaluation committee of INRIA (until summer 2011),
  - nominated member of the Section 7 of the Comité National de la Recherche Scientifique,
  - member of the hiring committees of *chaires* at Université Paris Dauphine and Télécom Paris Sud (president),
  - INRIA representative in the Scientific Directorate of the International Computer Science Meeting Center in Dagstuhl,
  - delegate for the organization of conferences at INRIA Nancy Grand-Est,
  - program committees ICFEM, SBMF, SEFM and SSS conferences, ATE, AVoCS, ICFEM, Refinement workshops, steering committees of AVoCS and IFM,
  - expert for the French Agence Nationale de la Recherche (ANR), AERES, the German DAAD, and the Canadian NSERC, and
  - PhD committees at Aalto University, Finland (report) and at ENS Cachan Bretagne (examiner).

9.2. Teaching

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

- Pascal Fontaine was head of an undergraduate program (Licence Miage) at Nancy 2 University in the academic year 2010/11.
- 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.
- Marie Duflot-Kremer and Stephan Merz taught a course on algorithmic verification in the Master's program in Nancy.

The following PhD theses were successfully defended in 2011 or are currently in preparation:

Diego Caminha Barbosa de Oliveira: *Fragments de l'arithmétique dans une combinaison de procédures de décision*, defended on 14 march 2011, supervised by Pascal Fontaine and Stephan Merz

Cristián Rosa: *Performance and Correctness Assessment of Distributed Systems*, defended on 24 october 2011, supervised by Stephan Merz and Martin Quinson (of team AlGorille)

Sabina Akhtar: *High-Level Language for Modeling Distributed Algorithms*, since 09/2008, supervised by Stephan Merz

Henri Debrat: *Vérification formelle d'algorithmes répartis avec erreurs byzantines*, since 10/2009, supervised by Bernadette Charron-Bost (CNRS & LIX) and Stephan Merz

Tianxiang Lu: *Formal Verification of a Peer-to-Peer Algorithm*, since 05/2009, supervised by Stephan Merz and Christoph Weidenbach of MPI-INF, Saarbrücken

Hernán-Pablo Vanzetto: *Model Construction for TLA$^+$ formulas*, since 10/2010, supervised by Kaustuv Chaudhuri (of INRIA Saclay) and Stephan Merz

10. Bibliography

Major publications by the team in recent years

[1] T. BOUTON, D. CAMINHA BARBOSA 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] K. CHAUDHURI, D. DOLIGEZ, L. LAMPORT, S. MERZ. *Verifying Safety Properties With the TLA+ Proof System*, in "Fifth Intl. Joint Conf. Automated Reasoning (IJCAR 2010)", Edinburgh, UK, J. GIESL, R. HÄHNLE (editors), LNCS, Springer, 2010, vol. 6173, p. 142–148 [*DOI :* 10.1007/978-3-642-14203-1_12], http://hal.inria.fr/inria-00534821/en.

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

[5] 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.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[6] D. CAMINHA BARBOSA DE OLIVEIRA. *Fragments de l'arithmétique dans une combinaison de procédures de décision*, Université Nancy II, March 2011, http://hal.inria.fr/tel-00578254/en.

[7] C. ROSA. *Performance and Correctness Assessment of Distributed Systems*, Université Henri Poincaré Nancy I, October 2011.

International Conferences with Proceedings

[8] C. ARECES, P. FONTAINE. *Combining theories: the Ackerman and Guarded Fragments*, in "8th International Symposium Frontiers of Combining Systems - FroCoS 2011", Saarbrücken, Germany, C. TINELLI, V. SOFRONIE-STOKKERMANS (editors), Lecture Notes in Computer Science, Springer Verlag, 2011, vol. 6989, p. 40–54 [*DOI :* 10.1007/978-3-642-24364-6_4], http://hal.inria.fr/hal-00642529/en.

[9] B. CHARRON-BOST, H. DEBRAT, S. MERZ. *Formal Verification of Consensus Algorithms Tolerating Malicious Faults*, in "13th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2011)", Grenoble, France, X. DÉFAGO, F. PETIT, V. VILLAIN (editors), Lecture Notes in Computer Science, Springer, October 2011, vol. 6976, p. 120-134 [*DOI :* 10.1007/978-3-642-24550-3_11], http://hal.inria.fr/hal-00639048/en.

[10] D. DÉHARBE, P. FONTAINE, S. MERZ, B. WOLTZENLOGEL PALEO. *Exploiting Symmetry in SMT Problems*, in "International Conference on Automated Deduction (CADE)", Wroclaw, Poland, N. BJØRNER, V.

SOFRONIE-STOKKERMANS (editors), Lecture Notes in Computer Science, Springer, August 2011, vol. 6803, p. 222-236 [*DOI :* 10.1007/978-3-642-22438-6_18], http://hal.inria.fr/inria-00617843/en.

[11] P. FONTAINE, S. MERZ, B. WOLTZENLOGEL PALEO. *Compression of Propositional Resolution Proofs via Partial Regularization*, in "23rd International Conference on Automated Deduction - CADE-23", Wroclaw, Poland, N. BJØRNER, V. SOFRONIE-STOKKERMANS (editors), Lecture Notes in Computer Science, Springer, August 2011, vol. 6803, p. 237-251 [*DOI :* 10.1007/978-3-642-22438-6_19], http://hal.inria.fr/inria-00617846/en.

[12] S. MERZ, T. LU, C. WEIDENBACH. *Towards Verification of the Pastry Protocol using TLA+*, in "31st IFIP International Conference on Formal Techniques for Networked and Distributed Systems", Reykjavik, Iceland, R. BRUNI, J. DINGEL (editors), June 2011, vol. 6722, http://hal.inria.fr/inria-00593523/en.

[13] S. MERZ, M. QUINSON, C. ROSA. *SimGrid MC: Verification Support for a Multi-API Simulation Platform*, in "31st IFIP International Conference on Formal Techniques for Networked and Distributed Systems", Reykjavik, Iceland, R. BRUNI, J. DINGEL (editors), Lecture Notes in Computer Science, Springer, June 2011, vol. 6722, p. 274-288, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-21461-5_18], http://hal.inria.fr/inria-00593505/en.

[14] D. MÉRY, M. MOSBAH, M. TOUNSI. *Refinement-based Verification of Local Synchronization Algorithms*, in "17th International Symposium on Formal Methods", Limerick, Ireland, Lecture Notes in Computer Science, Springer, June 2011, http://hal.inria.fr/hal-00579252/en.

[15] D. MÉRY, N. K. SINGH. *Analysis of DSR Protocol in Event-B*, in "13th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2011)", Grenoble, France, X. DÉFAGO, F. PETIT, V. VILLAIN (editors), Springer Berlin / Heidelberg, October 2011, vol. 6976, p. 401-415, http://hal.inria.fr/inria-00637768/en.

[16] B. WOLTZENLOGEL PALEO. *Atomic Cut Introduction by Resolution: Proof Structuring and Compression*, in "Logic for Programming, Artificial Intelligence, and Reasoning", Dakar, Senegal, E. M. CLARKE, A. VORONKOV (editors), Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence, Springer, June 2011, 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.inria.fr/hal-00545473/en.

Conferences without Proceedings

[17] F. BESSON, P. FONTAINE, L. THÉRY. *A Flexible Proof Format for SMT: a Proposal*, in "Workshop on Proof eXchange for Theorem Proving (PxTP)", Wroclaw, Poland, August 2011, http://hal.inria.fr/hal-00642544/en.

[18] D. DÉHARBE, P. FONTAINE, B. WOLTZENLOGEL PALEO. *Quantifier Inference Rules for SMT proofs*, in "Workshop on Proof eXchange for Theorem Proving (PxTP)", Wroclaw, Poland, 2011, http://hal.inria.fr/hal-00642535/en.

[19] S. MERZ, H. VANZETTO. *Towards certification of TLA+ proof obligations with SMT solvers*, in "Workshop on Proof eXchange for Theorem Proving - PxTP 2011", Wroclaw, Poland, P. FONTAINE, A. STUMP (editors), July 2011, http://hal.inria.fr/hal-00645458/en.

Research Reports

[20] M. B. ANDRIAMIARINA, D. MÉRY. *Stepwise development of distributed vertex coloring algorithms*, LORIA, July 2011, http://hal.inria.fr/inria-00606254/en.

References in notes

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

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

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

[29] L. LAMPORT. *The PlusCal Algorithm Language*, in "6th Intl. Coll. Theoretical Aspects of Computing (ICTAC 2009)", Kuala Lumpur, Malaysia, M. LEUCKER, C. MORGAN (editors), Lecture Notes in Computer Science, Springer, 2009, vol. 5684, p. 36-60.

[30] 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.

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

[32] Y. MÉTIVIER, J. ROBSON, N. SAHEB-DJAHROMI, A. ZEMMARI. *An Analysis of an Optimal Bit Complexity Randomised Distributed Vertex Colouring Algorithm*, in "13th Intl. Conf. Principles of Distributed Systems (OPODIS 2009)", Nîmes, France, T. F. ABDELZAHER, M. RAYNAL, N. SANTORO (editors), LNCS, Springer, 2009, vol. 5923, p. 359-364.

[33] B. PARNO, J. R. LORCH, J. R. DOUCEUR, J. W. MICKENS, J. M. MCCUNE. *Memoir: Practical State Continuity for Protected Modules*, in "IEEE Symposium on Security and Privacy", Berkely, CA, USA, IEEE Computer Society, 2011, p. 379-394.