IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2012

Project-Team VERIDIS

Modeling and Verification of Distributed Algorithms and Systems

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

RESEARCH CENTER
Nancy - Grand Est

THEME
Programs, Verification and Proofs

Table of contents

Project-Team VERIDIS

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

VeriDis is a joint research group 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.

Creation of the Project-Team: July 01, 2012 .

1. Members

Research Scientists

Stephan Merz [Team leader, Senior Researcher, HdR]
Uwe Waldmann [Max-Planck-Institute for Informatics, Saarbrücken]
Christoph Weidenbach [Team leader, Max-Planck-Institute for Informatics, Saarbrücken, HdR]

Faculty Members

Marie Duflot-Kremer [Associate Professor, Université de Lorraine, maternity leave 07-12/2012]
Pascal Fontaine [Associate Professor, on leave from Université de Lorraine since 09/2011]
Dominique Méry [Professor, Université de Lorraine, HdR]

External Collaborator

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

Engineer

Pablo Federico Dobal [since 09/2012]

PhD Students

Sabina Akhtar [Université de Lorraine, thesis defended on May 10, 2012]
Manamiary Andriamiarina [Université de Lorraine, since 10/2010]
Henri Debrat [Université de Lorraine, since 10/2009, joint supervision with Bernadette Charron-Bost]
Tianxiang Lu [Universität des Saarlandes and Université de Lorraine, since 05/2009]
Hernán-Pablo Vanzetto [Université de Lorraine, since 10/2010, joint supervision with Kaustuv Chaudhuri]
Arnaud Fietzke [Universität des Saarlandes]
Evgeny Kruglov [Universität des Saarlandes]
Daniel Wand [Universität des Saarlandes]

Post-Doctoral Fellow

Thomas Sturm [Max-Planck-Institute for Informatics, Saarbrücken]

Administrative Assistant

Sophie Drouot [shared with teams Bigs, Score, Tosca]

Others

Rodrigo Castaño [Universidad de Buenos Aires, Argentina, student intern 09-12/2012]
Simon Halfon [ENS Cachan, student intern 06-08/2012]

2. Overall Objectives

2.1. Overall Objectives

The VeriDis project team includes members of the MOSEL team of LORIA, the computer science laboratory in Nancy, and members of the Automation of Logic Research Group at Max-Planck Institute for Informatics (MPII) in Saarbrücken. It is headed by Stephan Merz and Christoph Weidenbach. VeriDis was created in 2010 as a local team of Inria Nancy Grand-Est. After a positive evaluation of the project proposal in the spring of 2011, the team was officially created in July 2012.

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

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

VeriDis proposes to exploit and further develop automation in system verification, and to apply its techniques within the context of concurrent and distributed algorithms, which are by now ubiquitous and whose verification is a big challenge. Concurrency problems are central to the development and verification of programs for multi- and many-core architectures, and distributed computation underlies the paradigms of grid and cloud computing. 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 potential of distributed systems for increased resilience to component failures makes them attractive in many contexts, but also makes formal verification important and challenging. We aim to move current research in this area on to a new level of productivity and quality. To give a concrete example: today 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.

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 [7], one of the leading automated theorem provers for first-order logic based on the superposition calculus [33]. Recent extensions to the system include the integration of dedicated reasoning procedures for specific theories, such as linear arithmetic [44], [31], that are ubiquitous in the verification of systems and algorithms. The group also studies general frameworks for the combination of theories such as the locality principle [45] and automated reasoning mechanisms these induce.

The VeriDis members from Nancy develop veriT [1], an SMT (Satisfiability Modulo Theories [35]) solver that combines decision procedures for different fragments of first-order logic and that integrates an automatic theorem prover for full first-order logic. The veriT solver is designed to produce detailed proofs; this makes it particularly suitable as a component of a robust cooperation of deduction tools.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction. Members of VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we participate in a project at the joint MSR-Inria Centre in Saclay on the development of methods and tools for the formal proof of TLA$^+$ [41] specifications. Our prover relies on a declarative proof language and includes several automatic backends [3].

3.2. Methodology of proved system development

Powerful theorem provers are not a panacea for system verification: they support sound methodologies for modeling and verifying systems. In this respect, members of VeriDis have gained expertise and recognition in making contributions to formal methods for concurrent and distributed algorithms and systems [2], [6], and in applying them to concrete use cases. In particular, the concept of *refinement* [30], [34], [43] in state-based modeling formalisms is central to our approach. Its basic idea is to derive an algorithm or implementation by providing a series of models, starting from a high-level description that precisely states the problem, and gradually adding details in intermediate models. An important goal in designing such methods is to reduce the number of generated proof obligations and/or support their proof by automatic tools. This requires taking into account specific characteristics of certain classes of systems and tailoring the model to concrete computational models. Our research in this area is supported by carrying out case studies for academic and industrial developments. This activity benefits from and influences the development of our proof tools.

Our vision for the integration of our expertise can be resumed as follows. Based on our experience and related work on specification languages, logical frameworks, and automatic theorem proving tools, we develop an approach that is suited for specification, interactive theorem proving, and for eventual automated analysis and verification, possibly through appropriate translation methods. While specifications are developed by users inside our framework, they are analyzed for errors by our SMT based verification tools. Eventually, properties are proved by a combination of interactive and automatic theorem proving tools, 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. We are in particular working on modeling and verifying medical devices that require closed-loop models of both the system and its environment.

5. Software

5.1. The veriT solver

Participants: Rodrigo Castaño, David Déharbe, Pablo Federico Dobal, Pascal Fontaine [correspondent].

The veriT solver is an SMT (Satisfiability Modulo Theories) solver developed in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brazil. The solver can handle large quantifier-free formulas containing uninterpreted predicates and functions, and arithmetic 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.

Efforts in 2012 have been focused on efficiency, with various improvements and the redesign of the core solver. A preliminary prototype integrating Redlog for handling non-linear arithmetic showed encouraging results. Short term future works include improving the design, adding full support for non-linear arithmetic, and increasing 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. In 2012, we presented at ABZ [16] a plugin for Rodin using SMT solvers (and notably veriT) to discharge B proof obligations: on a large repository of industrial and academic cases, this SMT-based plugin decreased by 75% the number of proof obligations requiring human interactions, compared to the original B prover. See also section 8.1 for our work within the DeCert project.

For helping development within and around veriT, Pablo Federico Dobal has been hired for two years starting September 2012 as a young engineer supported by the Inria ADT program.

5.2. The TLA+ proof system

Participants: Stephan Merz [correspondent], 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$^+$ 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, and a backend for interfacing with SMT solvers. The SMT backend has been improved significantly in 2012 and is now considered by users as the most useful backend prover for system verification. Version 1.0 of TLAPS was released in January 2012, followed by version 1.1 in November, and the system was presented at the conference FM 2012 [15].

6. New Results

6.1. Automated and Interactive Theorem Proving

6.1.1. Combination of decision procedures

Participants: Pascal Fontaine, Simon Halfon, Stephan Merz, Christoph Weidenbach.

SMT solvers, combination, decision procedures, theorem proving

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 generalized this result to the decidable first-order class of monadic predicate logic, and to the two-variable fragment. In subsequent joint work 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. In 2012, we considered, in the same spirit, the combination of theories that are not necessarily decidable [18]. In particular, we considered combinations of decision procedures and refutationally complete semi-decision procedures, as well as black-box combinations of different refutationally complete theorem provers, together with finite model finders. These results in particular yield theoretical foundations for how FOL provers can be combined with SMT techniques in a black-box style of integration.

6.1.2. Using symmetries in SMT

Participants: Pascal Fontaine, Stephan Merz.

theorem proving, SMT solvers, decision procedures, symmetry

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

In 2012, we designed a more general approach, based on graph isomorphism, for symmetry detection in the SMT context. Experimental analysis indicates that many formulas from the SMT-LIB repository exhibit symmetries that are left unexploited by the previous techniques. Finding new techniques to exploit these is the subject of ongoing work with the University of Cordoba in Argentina; we expect that breaking those symmetries will yield again some significative efficiency improvement.

6.1.3. Encoding TLA+ proof obligations for SMT solvers

Participants: Stephan Merz, Hernán-Pablo Vanzetto.

system verification, SMT solving, TLA

The TLA$^+$ proof system TLAPS (see section 5.2) is being developed within a project at the MSR-Inria Joint Centre to which we contribute. Proof obligations that arise during the verification of typical TLA$^+$ specifications require reasoning about the principal TLA$^+$ data structures such as sets, functions, arithmetic, tuples, and records. None of the backend provers present in the initial versions of TLAPS was able to reason effectively about steps involving several of these features, and in 2011 we started developing an improved backend for translating TLA$^+$ proof obligations to SMT-Lib, the generic input language of SMT solvers. The main challenge was to design a sound translation from untyped TLA$^+$ to the multi-sorted first-order logic that underlies SMT-Lib, and our original proposal was based on deriving type assignments to TLA$^+$ expressions in a custom type system useful for SMT-Lib. This approach sometimes failed to derive types for subexpressions or required stronger typing assumptions than those required by the semantics of untyped TLA$^+$.

In 2012, based on a suggestion by Ken McMillan, we investigated a different approach whose main idea is to embed SMT sorts such as integers in the global universe of TLA$^+$ values, and to axiomatically define operations such as addition or multiplication on the image of that embedding. This approach effectively delegates type inference to the SMT solver and can therefore handle arbitrary TLA$^+$ expressions. However, it generates many quantified background axioms that may render SMT solvers ineffective, and we developed powerful pre-processing techniques for replacing quantified axioms by their required ground instances. The SMT backend in the current release of TLAPS is based on a hybrid approach to translation, where type inference is used whenever possible in order to obtain simpler SMT input. The two translation techniques have been published in 2012 [19], [20], and they have been validated over many case studies in TLAPS. For example, it enables proving the correctness of simple mutual-exclusion algorithms essentially without user interaction, and of the Paxos consensus algorithm in just 130 interactions, whereas a previous proof attempt using the traditional backend provers was unsuccessful.

6.1.4. Compression of SMT proofs

Participants: Pascal Fontaine, Stephan Merz.

theorem proving, SMT solvers, decision procedures, combination of decision procedures

Integrating an SMT solver in a certified environment such as TLAPS or 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 size 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, but we proposed practical and efficient algorithms for more restricted compression techniques. We continue to develop this line of work with our partners at TU Wien.

6.1.5. Augmenting the Expressiveness of Spass

Participants: Evgeny Kruglov, Arnaud Fietzke, Daniel Wand, Christoph Weidenbach.

automated theorem proving, superposition, linear arithmetic, proof assistants

In 2012 we focused on bridging the gap between the input logic of SPASS and more expressive logics as they are used by systems supporting full-fledged verification such as Isabelle and TLAPS. Main contributions were a specific version of an order-sorted language that can be eventually translated in a many-sorted logic. The latter is implemented in Spass in a prototypic way and first experiments showed significant improvements on proof obligations out of Isabelle/HOL. Actually, the enhancements allowed Spass to become the most powerful automated theorem proving system supporting Isabelle [14]. We are currently working on a coupling with TLAPS (see section 5.2).

A second important branch is the integration of arithmetic into SPASS and the development of the respective hierarchic superposition calculus. In the past [31], [38] we experimented with a black box integration of LP solvers and Z3 to delegate arithmetic reasoning tasks. Now we started our own white box implementation for linear arithmetic and could achieve significant speed-ups. Our own reasoning procedure, dedicated to the

specific form of the arithmetic proof obligations generated by SPASS is 50 to 200 times faster than any black box integration [29]. On the calculus side we could prove hierarchic superposition modulo linear arithmetic to be a decision procedure for the ground case, thus strictly generalizing the DPLL(LA) set up, and to be a decision procedure [39], [40] for timed automata reachability and extensions thereof [17].

6.1.6. Verification of linear hybrid automata

Participant: Uwe Waldmann.

automated theorem proving, superposition, linear arithmetic, proof assistants

We propose an improved symbolic algorithm for the verification of linear hybrid automata with large discrete state spaces. Large discrete state spaces arise naturally in industrial hybrid systems, due to the need to represent discrete inputs, counters, sanity checkbits, possibly multiple concurrent state machines, system-degradation modes, and finite switching variables. To prove safety properties of such systems, it is necessary to combine techniques for analyzing a complex dynamic behaviour with state space exploration methods that can deal with hundreds of discrete variables. In our approach, we represent both the discrete part and the continuous part of the hybrid state space symbolically using a variant of AIGs (And-Inverter-Graphs). Key components of our method are redundancy elimination (to maintain a compact symbolic representation by deleting superfluous linear constraints) and constraint minimization (exploiting the fact that states already reached in previous iterations of the model-checking algorithm can be interpreted as "don't cares" in later steps). A journal article describing the technique appeared in Science of Computer Programming [9].

6.2. Proved development of algorithms and systems

6.2.1. Incremental development of distributed algorithms

Participants: Dominique Méry, Manamiary Andriamiarina.

distributed algorithms, refinement, verification, distributed protocols

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 was initially (until 2010) carried out within the ANR project RIMEL, in joint work with Mohammed Mosbah and Mohammed Tounsi from the LABRI laboratory, 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.

More precisely, we show how state-based models can be developed for specific problems and how they can be simply reused by controlling the composition of state-based models through the refinement relationship. Consequently, we obtain a redevelopment of existing distributed algorithms in the *correct-by-construction* approach, and a framework for deriving new distributed algorithms (by integrating models) whose correctness is ensured by construction. Traditionally, distributed algorithms are supposed to run on a fixed network, whereas we consider a network with a changing topology. We have illustrated our methodology with the study of the protocol ANYCAST RP.

The contribution is related to the development of proof-based patterns providing effective help to the developer of formal models of applications, such as dynamic routing or the snapshot problem [13]. In fact, we have developed patterns for simplifying the development of distributed systems using refinement. The applicability of a pattern for routing has been reapplied to the development of a network on chip [12] with our partners of the French-Algerian cooperation described in section 8.3.

6.2.2. Modeling and verifying the Pastry routing protocol

Participants: Tianxiang Lu, Stephan Merz, Christoph Weidenbach.

distributed hash table, peer-to-peer protocol, Pastry, model checking, theorem proving

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 [36] for maintaining a distributed hash table in a peer-to-peer network. As part of his PhD work, 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.

As a first step towards proving correctness of the Pastry routing protocol, we identified in 2011 a number of candidate invariants and formally proved in TLAPS (see section 5.2) that these implied the high-level correctness property. In 2012, we consolidated these invariants and proved them correct for our model under the strong assumption that no node ever leaves the network, and the minor assumption that any active node can at any time only allow one new node to join the network. It is still not clear at the moment to which extent nodes can be allowed to leave the network without breaking the virtual ring maintained by Pastry. The invariant proofs contain almost 15000 interactions and constitutes the largest case study carried out so far using TLAPS. We have more recently been able to obtain better automation using the new SMT backend (see section 6.1). The proof was presented at the TLA workshop of FM 2012 [23].

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

Participants: Henri Debrat, Stephan Merz.

theorem proving, distributed algorithms, round-based computation, Byzantine failures

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 [37], and have previously established a reduction theorem that allows to pretend that nodes operate synchronously.

In 2012, we have consolidated our formal proofs in Isabelle/HOL. In particular, we have finished the formal proof of the reduction theorem within Isabelle, produced a generic encoding of the Heard-Of model as a locale in Isabelle/HOL, and used this representation for verifying six different Consensus algorithms: three algorithms tolerating benign failures and three others designed for malicious failures, such as corrupted values. Our Isabelle theories have been published at the Archive of Formal Proofs [27]. The proof of the reduction theorem required formalizing the notion of stuttering invariance, which can be of independent interest and that has also been accepted at the Archive of Formal Proofs [28].

As a significant extension of this work, we have studied the formal verification of probabilistic Consensus algorithms in the Heard-Of model, in particular the Ben-Or algorithm.

6.2.4. Model checking within SimGrid

Participants: Marie Duflot-Kremer, Stephan Merz.

model checking, distributed algorithms, message passing, communication primitives, partial-order reduction

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. As part of the thesis work of Cristián Rosa (defended in 2011), a stateless model checker was implemented within the SimGrid platform that can be used to verify safety properties of distributed C programs that communicate by message passing. The ongoing thesis of Marion Guthmuller builds upon this work and aims to extend it for verifying certain liveness properties. This requires rethinking the stateless design, as well

as adapting the dynamic partial-order reduction algorithm that is essential to limiting the part of the state space that must actually be explored.

6.2.5. Modeling Medical Devices

Participant: Dominique Méry.

Formal modelling techniques and tools have attained sufficient maturity for formalizing highly critical systems in view of improving their quality and reliability, and the development of such methods has attracted the interest of industrial partners and academic research institutions. Building high quality and zero-defect medical software-based devices is a particular domain where formal modelling techniques can be applied effectively. In [21], we present a methodology for developing critical systems from requirement analysis to automatic code generation based on a standard safety assessment approach. This methodology combines refinement, proof, model checking, and animation, and ultimately can automatically generate source code. This approach is intended to contribute to further the use of formal techniques for developing critical systems with high integrity and to verify complex properties. An assessment of the proposed methodology is given through developing a standard case study: the cardiac pacemaker.

Medical devices are very prone to showing unexpected system behaviour in operation when traditional methods are used for system testing. Device-related problems have been responsible for a large number of serious injuries. Officials of the US Food and Drug Administration (FDA) found that many deaths and injuries related to these devices are caused by flaws in product design and engineering. Cardiac pacemakers and implantable cardioverter-defibrillators (ICDs) are among the most critical medical devices and require closed-loop modelling (integrated system and environment modelling) for verification purposes before obtaining a certificate from the certification bodies. In [24] we present a methodology for modelling a biological system, such as the heart. The heart model is based mainly on electrocardiography analysis, which provides a model at the cellular level. Combining this environment model with a formal model of the pacemaker, we obtain a closed-loop model over which the overall correctness can be verified.

Clinical guidelines systematically assist practitioners in providing appropriate health care in specific clinical circumstances. Today, a significant number of guidelines and protocols are lacking in quality. Indeed, ambiguity and incompleteness are likely anomalies in medical practice. In [25] we use the Event-B modeling language to represent guidelines for subsequent validation. Our main contributions are: to apply mathematical formal techniques to evaluate real-life medical protocols for quality improvement, to derive verification proofs for the protocol and properties according to medical experts, and to publicize the potential of this approach. An assessment of the proposed approach is given through a case study, relative to a real-life reference protocol concerning ECG interpretation, for which we uncovered several anomalies.

Finally, we propose a refinement-based methodology [10] for complex medical systems design, which possesses the required key features. A refinement-based combined approach of formal verification, model validation using a model-checker and refinement chart is proposed in this methodology for designing a high-confidence medical device. Furthermore, we show the effectiveness of this methodology for the design of a cardiac pacemaker system.

6.2.6. Fundamentals of Network Calculus in Isabelle/HOL

Participant: Stephan Merz.

networked systems, min-plus algebra, formal proof

The design of networked and embedded systems has traditionally been accompanied by formal methods for design and analysis. Network Calculus [42] is a well-established theory, based on the $(\min, +)$ dioid, that is designed for computing delay and memory bounds in networks. The theory is supported by several commercial and open-source tools and has been used in major industrial applications, such as the design and certification of the Airbus A380 AFDX backbone. Nevertheless, it is difficult for certification authorities to assess the correctness of the computations carried out by the tools supporting Network Calculus, and we propose the use of *result certification* techniques for increasing the confidence in the Network Calculus toolchain. In joint work with Marc Boyer from ONERA in Toulouse, and with Loïc Fejoz and Nicolas Navet from the

RealTime at Work (RTaW) company, we have supervised the master thesis of Etienne Mabille to evaluate the feasibility of the approach. Parts of the theory underlying Network Calculus were formalized in the proof assistant Isabelle/HOL, and this encoding was used to formally derive theorems that underly the computation of bounds in network servers. The Network Calculus tool produced by RTaW was instrumented to generate traces of its computation, and the correctness of simple systems could in this way be certified by Isabelle. A publication of this work is in preparation, and we intend to continue and extend it in a future joint project.

6.2.7. Bounding message length in attacks against security protocols

Participant: Marie Duflot-Kremer.

security protocols, verification

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 [32] that, under a syntactic and reasonable condition of "well-formedness" on the protocol, we can get rid of the infinitely branching part. Following this conference publication, we are preparing a journal version of this result extending the set of security properties to which the result is applicable, in particular including authentication properties.

6.2.8. Evaluating and verifying probabilistic systems

Participant: Marie Duflot-Kremer.

verification, probabilistic systems, performance evaluation

Since its introduction in the 1980s, model checking has become a prominent technique for the verification of complex systems. The aim was to decide whether or not a system was fulfilling its specification. With the rise of probabilistic systems, new techniques have been designed to verify this new type of systems, and appropriate logics have been proposed to describe more subtle properties to be verified. However, some characteristics of such systems cannot fall in the field of model checking. The aim is thus not to tell wether a property is satisfied but how well the system performs with respect to a certain measure. Together with researchers from ENS de Cachan and University Paris Est Créteil we have designed a statistical tool made to tackle both performance and verification issues. Following several conference talks, a journal paper is currently written to present both the approach as well as application to a concrete case study: flexible manufacturing systems.

7. Bilateral Contracts and Grants with Industry

7.1. 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 6.1). In particular, the project funds the PhD thesis of Hernán Vanzetto.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

Participants: Pascal Fontaine, Stephan Merz.

The DeCert (Deduction and Certification) project has been funded by ANR from 2009 to 2012 within its "Domaines émergents" program. It was 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 has been 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).

8.1.2. Inria Development Action VeriT

Participants: Pablo Federico Dobal, Pascal Fontaine.

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

8.2. European Initiatives

8.2.1. Cooperation with TU Wien, Austria

Participants: Pascal Fontaine, Stephan Merz.

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

A first workshop of one week took place in Vienna in spring, and gathered around 15 people, including Pascal Fontaine and Stephan Merz as well as a student from TU Graz. A second one-week workshop was organized in Nancy in the fall, with 12 participants including 5 researchers from Vienna, and one student from Univ. Paul Sabatier, Toulouse. The web page gives more information on this project.

8.3. International Initiatives

8.3.1. Participation In International Programs

8.3.1.1. Cooperation with Córdoba, Argentina

Participants: Pascal Fontaine, Stephan Merz.

This cooperation with the team of Carlos Areces (formerly a researcher at Inria Nancy) at the University of Córdoba is along two axes. First, we study symmetries for automated reasoning (and SMT) as a means to reduce the search space and improve efficiency. Second, we investigate automated reasoning techniques (and more specifically SMT) for modal logics and similar fragments of first-order logic. The cooperation is funded within the context of the IRSES project MEALS coordinated for Inria by Catuscia Palamidessi (Saclay).

Two PhD students from Córdoba visited Inria Nancy in Summer 2012: Ezequiel Orbe for two weeks, and Raul Fervari for one month. Carlos Areces also came to Nancy for two weeks. Pascal Fontaine and Stephan Merz visited Argentina in November where they spent two weeks in Córdoba working on the above subjects, and one week visiting our contacts at the universities of Rosario and Buenos Aires.

The team has a long term relationship with the Universities of Córdoba, Rosario and Buenos Aires, with frequent exchanges of students. One Internship student in 2012 was from Buenos Aires, and the newly recruited engineer is from Rosario.

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

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

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

8.3.1.3. Cooperation with Tiaret University

Participants: Dominique Méry, Stephan Merz.

Mostapha Belardi (Université Ibn Khaldoun de Tiaret), Camel Tanougast (LICM, Université de Lorraine), 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, which funded a visit of Mostapha Belardi and an internship of Hayat Daoud in 2012. The work led to the design of a model for a network on chip proposed by our partners from LICM. A short presentation has been published in a local workshop.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

David Déharbe from Universidade Federal do Rio Grande de Norte, Brazil, visited VeriDis from July 9 to July 27 and from October 15 to October 26 in the context of the Inria-CNPq project SMT-SAVeS. The work resulted in several improvements of the veriT solver.

Thomas Sturm, from MPI für Informatik, and Ulrich Loup and Florian Corzilius, from RWTH Aachen, visited VeriDis from October 22nd to 26th, in the context of the ADT veriT for discussing techniques for non-linear arithmetic in SMT solving.

8.4.2. International Internships

- Rodrigo Castaño (from Sep 2012 until Dec 2012)
  - Subject: Methods for efficient SMT solving
  - Institution: University of Buenos Aires (Argentina)

9. Dissemination

9.1. Scientific Animation

- Pascal Fontaine co-chaired the program committee of PAAR 2012 and SMT 2012. He served on the program committee of PxTP 2012. He has been ex-officio member of the SMT Steering Committee starting September 2011 until August 2012, and he is now an elected member for two years starting September 2012.
- Dominique Méry is
  - a member of the IFIP Working Group 1.3 on *Foundations of System Specification*,
  - head of the Doctoral School IAEM Lorraine for the University of Lorraine,
  - head of the Formal Methods department of the LORIA laboratory,

  - an expert for the French Ministry of Education (DS9),
  - an expert for the French Agence Nationale de la Recherche (ANR) and AERES.
  - the director of international affairs at ESIAL Nancy, and
  - the president of the APCB association.
  - He served on the program committees of DS-Event-B, FHIES, FM (co-chair), ICECCS, ICFEM, iFM, MEDI, and TASE.
- The academic duties of Stephan Merz include:
  - member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*,
  - nominated member of the Section 7 of the Comité National de la Recherche Scientifique (until the summer of 2012),
  - Inria representative in the Scientific Directorate of the International Computer Science Meeting Center in Dagstuhl,
  - delegate for the organization of conferences at Inria Nancy Grand-Est,
  - co-head of the PhD committee for computer science in Lorraine (since December 2012),
  - program committees of ICFEM, iFM, SAC, SBMF, and SEFM conferences, AVoCS (co-chair), ATX, and TLA (co-chair) workshops, steering committee of AVoCS,
  - co-organizer of the VTSA summer school between Nancy, Saarbrücken, Luxembourg, and Liège,
  - expert for the French Agence Nationale de la Recherche (ANR), AERES, German DFG, European ERC, and Canadian NSERC.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

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

- Dominique Méry gave courses in the Master program in Nancy on: formal system engineering, modeling and verification of systems, theoretical computer science, development of software systems, distributed algorithms.
- Stephan Merz taught a course on algorithmic verification in the Master program in Nancy (30 hours) and a course on modal logic in the undergraduate curriculum on cognitive and computer science (30 hours).

9.2.2. Supervision

PhD: Sabina Akhtar, Verification of Distributed Algorithms using PlusCal-2, Université de Lorraine, May 2012, supervised by Stephan Merz;

PhD in progress: Manamiary Andriamiarina, Refinement Techniques for Distributed Algorithms, since 10/2010, supervised by Dominique Méry;

PhD in progress: Henri Debrat, Formal Verification of Fault-Tolerant Distributed Algorithms, since 10/2009, supervised by Bernadette Charron-Bost and Stephan Merz;

PhD in progress: Tianxiang Lu, Verification of the Pastry Routing Protocol, since 05/2009, supervised by Stephan Merz and Christoph Weidenbach;

PhD in progress: Hernán Vanzetto, SMT Techniques for TLA$^+$ Proof Obligations, since 10/2010, supervised by Kaustuv Chaudhuri and Stephan Merz.

9.2.3. Juries

Dominique Méry wrote reports on the following PhD theses.

- Anton Tarasyuk: *Formal Development and Quantitative Verification of Dependable Systems*, University of Turku;
- Jean-Charles Chaudemar: *Etude des architectures de sécurité de systèmes autonomes – Formalisation et évaluation en Event B*, Université de Toulouse;
- Damien Imbs: *Calculabilité et conditions de progression des objets partagés en présence de défaillances*, Université de Rennes;
- Vincent Filou: *Une étude formelle de la théorie des calculs locaux à l'aide de l'assistant de preuve Coq*, Université de Bordeaux.

Stephan Merz wrote reports on the following PhD and habilitation theses:

- Francesco Bongiovanni: *Design, Formalization and implementation of overlay networks; application to RDF data storage*, Université de Nice-Sophia Antipolis;
- Abderrahmane Feliachi: *Semantics Based Testing for Circus*, Université Paris-Sud;
- Sylvain Conchon: *SMT Techniques and their Applications*, Université Paris-Sud.

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

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

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

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

Publications of the year

Doctoral Dissertations and Habilitation Theses

[8] S. AKHTAR. *Verification of Distributed Algorithms using PlusCal-2*, Université de Lorraine, May 2012.

Articles in International Peer-Reviewed Journals

[9] W. DAMM, H. DIERKS, S. DISCH, W. HAGEMANN, F. PIGORSCH, C. SCHOLL, U. WALDMANN, B. WIRTZ. *Exact and fully symbolic verification of linear hybrid automata with large discrete state spaces*, in "Science of Computer Programming", September 2012, vol. 77, n^{o} 10-11, p. 1122-1150, http://hal.inria.fr/hal-00760387.

[10] D. MÉRY, N. K. SINGH. *Formal Specification of Medical Systems by Proof-Based Refinement*, in "ACM Transactions in Embedded Computing Systems", January 2012, http://hal.inria.fr/inria-00637756.

International Conferences with Proceedings

[11] Y. AIT AMEUR, D. MÉRY. *Handling Heterogeneity in Formal Developments of Hardware and Software Systems*, in "Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies - 5th International Symposium, ISoLA 2012", Heraklion, Greece, T. MARGARIA, B. STEFFEN (editors), Lecture Notes in Computer Science, Springer, October 2012, vol. 7610, p. 327-328, http://hal.inria.fr/hal-00743810.

[12] M. B. ANDRIAMIARINA, H. DAOUD, M. BELARBI, D. MÉRY, C. TANOUGAST. *Formal Verification of Fault Tolerant NoC-based Architecture*, in "First International Workshop on Mathematics and Computer Science (IWMCS2012)", Tiaret, Algeria, December 2012, http://hal.inria.fr/hal-00763092.

[13] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Revisiting Snapshot Algorithms by Refinement-based Techniques*, in "PDCAT 2012 : The Thirteenth International Conference on Parallel and Distributed Computing, Applications and Technologies", Beijing, China, 2012, http://hal.inria.fr/hal-00734131.

[14] J. BLANCHETTE, A. POPESCU, D. WAND, C. WEIDENBACH. *More SPASS with Isabelle – Superposition with Hard Sorts and Configurable Simplification*, in "Interactive Theorem Proving (ITP 2012)", Princeton, New Jersey, United States, L. BERINGER, A. FELTY (editors), LNCS, Springer, 2012, vol. 7406, p. 345-360, http://hal.inria.fr/hal-00760392.

[15] D. COUSINEAU, D. DOLIGEZ, L. LAMPORT, S. MERZ, D. RICKETTS, H. VANZETTO. *TLA+ Proofs*, in "18th International Symposium On Formal Methods - FM 2012", Paris, France, D. GIANNAKOPOULOU, D. MÉRY (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7436, p. 147-154, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-32759-9_14], http://hal.inria.fr/hal-00726631.

[16] D. DÉHARBE, P. FONTAINE, Y. GUYOT, L. VOISIN. *SMT solvers for Rodin*, in "ABZ - Third International Conference on Abstract State Machines, Alloy, B, VDM, and Z - 2012", Pisa, Italy, J. DERRICK, J. A. FITZGERALD, S. GNESI, S. KHURSHID, M. LEUSCHEL, S. REEVES, E. RICCOBENE (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7316, p. 194-207, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-30885-7_14], http://hal.inria.fr/hal-00747269.

[17] A. FIETZKE, E. KRUGLOV, C. WEIDENBACH. *Automatic Generation of Invariants for Circular Derivations in SUP(LA)*, in "18th International Conference on Logic for Programming, Artificial Intelligence and Reason-

ing", Mérida, Venezuela, Bolivarian Republic Of, N. BJØRNER, A. VORONKOV (editors), LNCS, Springer, 2012, vol. 7180, p. 197-211, http://hal.inria.fr/hal-00760398.

[18] P. FONTAINE, S. MERZ, C. WEIDENBACH. *Combination of disjoint theories: beyond decidability*, in "IJCAR - 6th International Joint Conference on Automated Reasoning - 2012", Manchester, United Kingdom, B. GRAMLICH, D. MILLER, U. SATTLER (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7364, p. 256-270, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-31365-3_21], http://hal.inria.fr/hal-00747271.

[19] S. MERZ, H. VANZETTO. *Automatic Verification Of TLA$^+$ Proof Obligations With SMT Solvers*, in "18th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18)", Mérida, Venezuela, Bolivarian Republic Of, N. BJØRNER, A. VORONKOV (editors), Lecture Notes in Computer Science, Springer, March 2012, vol. 7180, p. 289-303 [*DOI :* 10.1007/978-3-642-28717-6_23], http://hal.inria.fr/hal-00760570.

[20] S. MERZ, H. VANZETTO. *Harnessing SMT Solvers for TLA+ Proofs*, in "12th International Workshop on Automated Verification of Critical Systems (AVoCS 2012)", Bamberg, Germany, G. LÜTTGEN, S. MERZ (editors), ECEASST, EASST, December 2012, vol. 53, http://hal.inria.fr/hal-00760579.

[21] D. MÉRY, N. K. SINGH. *Critical systems development methodology using formal techniques*, in "3rd International Symposium on Information and Communication Technology - SoICT 2012", Ha Long, Viet Nam, ACM, August 2012, p. 3-12 [*DOI :* 10.1145/2350716.2350720], http://hal.inria.fr/hal-00747305.

Conferences without Proceedings

[22] D. COUSINEAU, D. DOLIGEZ, L. LAMPORT, S. MERZ, D. RICKETTS, H. VANZETTO. *TLA+ Proofs*, in "AI meets Formal Software Development", Dagstuhl, Germany, 2012, 16 p., http://hal.inria.fr/hal-00726632.

[23] T. LU, S. MERZ, C. WEIDENBACH. *Formal Verification Of Pastry Using TLA+*, in "International Workshop on the TLA+ Method and Tools", Paris, France, L. LAMPORT, S. MERZ (editors), August 2012, http://hal.inria.fr/hal-00768812.

Scientific Books (or Scientific Book chapters)

[24] D. MÉRY, N. K. SINGH. *Formalization of Heart Models Based on the Conduction of Electrical Impulses and Cellular Automata*, in "Foundations of Health Informatics Engineering and Systems", Z. LIU, A. WASSYNG (editors), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, vol. 7151, p. 140-159 [*DOI :* 10.1007/978-3-642-32355-3_9], http://hal.inria.fr/hal-00762821.

[25] D. MÉRY, N. K. SINGH. *Medical Protocol Diagnosis Using Formal Methods*, in "Foundations of Health Informatics Engineering and Systems", Z. LIU, A. WASSYNG (editors), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, vol. 7151, p. 1-20 [*DOI :* 10.1007/978-3-642-32355-3_1], http://hal.inria.fr/hal-00762822.

Books or Proceedings Editing

[26] D. GIANNAKOPOULOU, D. MÉRY (editors). *FM 2012: Formal Methods - 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings*, LNCS, Springer, August 2012, vol. 7436, 488, http://hal.inria.fr/hal-00743808.

Research Reports

[27] H. DEBRAT, S. MERZ. *Verifying Fault-Tolerant Distributed Algorithms in the Heard-Of Model*, Archive of Formal Proofs, July 2012, Published at http://afp.sourceforge.net/entries/Heard_Of.shtml, http://hal.inria.fr/hal-00760686.

[28] S. MERZ. *Stuttering Equivalence*, Archive of Formal Proofs, May 2012, Published at http://afp.sourceforge.net/entries/Stuttering_Equivalence.shtml, http://hal.inria.fr/hal-00760690.

Other Publications

[29] M. BROMBERGER. *Adapting the Simplex Algorithm for Superposition Modulo Linear Arithmetic*, Universität des Saarlandes, Saarbrücken, 2012, http://hal.inria.fr/hal-00760395.

References in notes

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

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] A. EGGERS, E. KRUGLOV, S. KUPFERSCHMID, K. SCHEIBLER, T. TEIGE, C. WEIDENBACH. *Superposition Modulo Non-linear Arithmetic*, in "Frontiers of Combining Systems, 8th International Symposium, FroCoS 2011, Saarbrücken, Germany, October 5-7, 2011. Proceedings", C. TINELLI, V. SOFRONIE-STOKKERMANS (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6989, p. 119-134.

[39] A. FIETZKE, C. WEIDENBACH. *Superposition as a decision procedure for timed automata*, in "MACIS 2011: Fourth Intl. Conf. Mathematical Aspects of Computer and Information Sciences", S. RATSCHAN (editor), 2011, p. 52–62.

[40] E. KRUGLOV, C. WEIDENBACH. *SUP(T) decides first-order logic fragment over ground theories*, in "MACIS 2011: Fourth Intl. Conf. Mathematical Aspects of Computer and Information Sciences", S. RATSCHAN (editor), 2011, p. 126–148.

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

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

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

[44] 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.

[45] 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.