INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Cassis

Combining approaches for the security of infinite state systems

Nancy - Grand Est

THEME SYM

Table of contents

1. Team

Head of project team
Michaël Rusinowitch [Research Director (DR), INRIA-LORIA, HdR]

Vice-head of project team
Olga Kouchnarenko [PR, Université Franche-Comté, LIFC, HdR]

Administrative assistant
Emmanuelle Deschamps

Permanent researchers
Serge Burckel [MC, U. de la Réunion, seconded to INRIA, from September 1]
Yannick Chevalier [MC, U. Paul Sabatier, Toulouse, seconded to INRIA, from September 1]
Véronique Cortier [CR, CNRS-LORIA]
Silvio Ranise [CR, INRIA-LORIA, in sabbatical stay, Univ. of Milan, from September 1]
Christophe Ringeissen [CR, INRIA-LORIA]
Mathieu Turuani [CR, INRIA-LORIA]

Faculty members
Fabrice Bouquet [MC, Université Franche-Comté, HdR]
Jean-François Couchot [PRAG, Université Franche-Comté, from September 1]
Frédéric Dadeau [MC, Université Franche-Comté, from September 1]
Alain Giorgetti [MC, Université Franche-Comté]
Pierre-Cyrille Héam [MC, Université Franche-Comté]
Abdessamad Imine [MC, Université Nancy 2, from September 1]
Laurent Vigneron [MC, Université Nancy 2]

PhD students
Thibaut Brocard [BDI-CNRS, LIFC, from October 1, until October 1, 2010]
Najah Chridi [MENRT, LORIA, until October 1, 2008]
Roméo Courbis [LIFC, from November 1, until November 1, 2010]
Stéphane Debricon [INTERREG, LIFC, from January 1, until January 1, 2010]
Heinrich Hoerdegen [MENRT, LORIA, thesis defended on November 29]
Adrien de Kermadec [VALMI, Co-tutelle LIFC and New-Zeland, from October 1, until October 1, 2010]
Vincent Pretre [INTERREG, LIFC]
Duc-Khanh Tran [MENRT, LORIA, until August 31, thesis defended on February 16]
Jérôme Voinot [LIFC, until November 1, 2009]
Eugen Zalinescu [MENRT, LORIA, thesis defended on December 17]
Daniele Zucchelli ["Cotutelle" between U. of Milan and U. Henri Poincaré, Nancy. Expected date of thesis defense: January 22, 2008]

Post-doctoral fellows
Stéphanie Delaune [Post-doctoral, RNTL POSÉ, LORIA, until September 30]
Mahat Khelfallat [Post-doctoral, RNTL DANOCOPS, LIFC, until August 31]
Enrica Nicolini [Ingénieur expert, QSL, LORIA, and Post-doctoral INRIA from December 1, until March 31, 2009]

Technical staff
Romain Pichon [Engineer, RNTL POSÉ, LIFC, From January 1 until September 30]

2. Overall Objectives

2.1. Background

Cassis is a joint project between the *Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA - UMR 7503)* and *Laboratoire d'Informatique de l'Université de ' Franche-Comté (LIFC - FRE 2661)*.

The objective of the project is to design and develop tools to verify the safety of systems with an infinite number of states. The analysis of such systems is based on a symbolic representation of sets of states in terms of formal languages or logical formulas. Safety is obtained via automatic proof, symbolic exploration of models or test generation. These validation methods are complementary. They rely on the study of accessibility problems and their reduction to constraint solving.

An originality of the project is its focus on infinite systems, parameterized or large scale, for which each technique taken separately shows its limits. This is the case for example with protocols operating on topologies of arbitrary size (ring networks), systems handling data structures of any size (sets), or whose control is infinite (automata communicating through an unbounded buffer). Ongoing or envisioned applications concern embedded software (e.g., smart cards, automotive controllers), cryptographic protocols (IKE, SET, TLS, Kerberos) designed to ensure trust in electronic transactions, and distributed systems.

The problem of validating or verifying reactive systems is crucial because of the increasing number of security-sensitive systems. The failure of these critical systems can have dramatic consequences since they may be embedded in vehicles components, or they control power stations or telecommunication networks. Beside obvious security issues, the reliability of products whose destination is millions of end-users has a tremendous economical impact.

There are several approaches to system verification: automated deduction, reachability analysis or model-checking, and testing. These approaches have different advantages and drawbacks. Automated deduction can address practical verification, however it remains complex to handle and requires a lot of expertise and guidance from the user. Model-checking is exhaustive but must face combinatorial explosion and becomes problematic with large-size or infinite systems. Testing is fundamental for validating requirements since it allows the discovery of many errors. However, it is almost never exhaustive and therefore only leads to partial solutions. Hence we believe that these approaches should not be considered as competing but as complementary.

The goal of our project is to contribute to new combinations of these three verification techniques in a framework that would apply them in an industrial context. In particular we expect some breakthrough in the infinite-state verification domain by joint applications of deductive, model-checking and testing techniques.

2.2. Context

For verifying the security of infinite state systems we rely on

- Different ways to express the safety, reachability or liveness properties of systems, linear-time or branching-time logics, and the application of abstraction or abstract interpretation.
- Test generation techniques.
- The modeling of systems by encoding states as words, terms or trees and by representing infinite sets of states by languages. To each of these structures corresponds appropriate action families, such as transductions or rewritings.

Our goal is to apply these different approaches for ensuring the security of industrial systems by providing adequate methods and tools. In more details we aim at the following contributions (see the continuous lines in Figure 1):

1. verification of abstract models derived from existing systems;
2. tests generation from the abstract model for validating the existing model;
3. cross-fertilization of the different validation techniques (deduction, model-checking, testing) by taking advantage of the complementary scopes and of their respective algorithmic contributions.

Let us mention that all these techniques comply with various development methodologies.

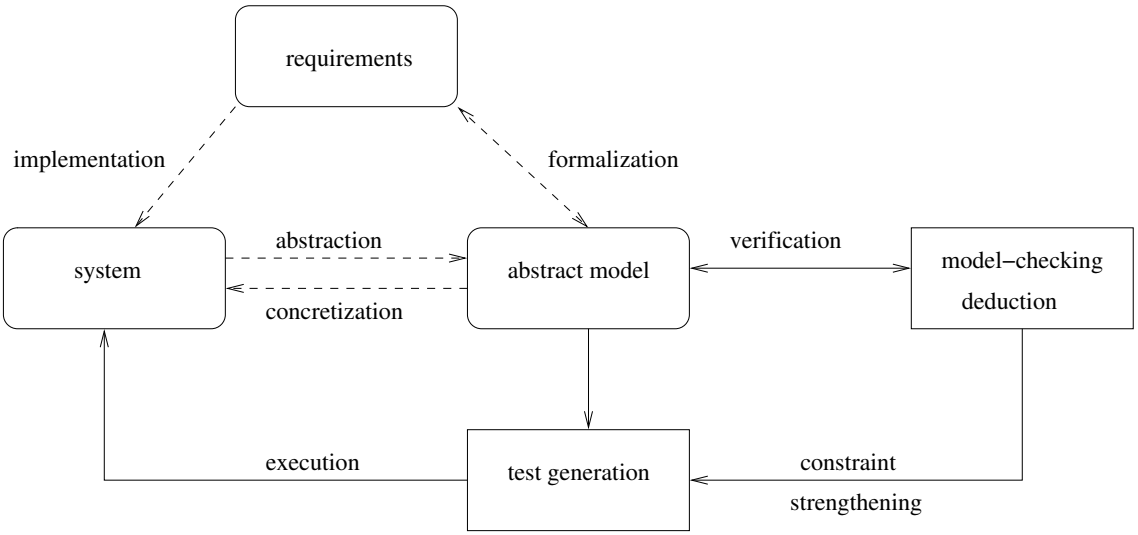

Figure 1. Software validation in Cassis

2.3. Challenge

Verifying the safety of infinite state systems is a challenge: nowadays algorithmic techniques only apply to very specific infinite state systems. On the other hand the deductive approaches are good candidates to capture infinite system safety verification but are difficult to bring into operation and require a deep expertise. A solution consists of integrating several verification methods by combining, for example, theorem-proving and model-checking.

The behavior of infinite states systems is expressed in the various models by composing or iterating actions. One of the main problems with algorithmic techniques is to compute the effect of these actions on the initial state. This computation is called *reachability analysis*. The verification of safety properties as well as the automatic generation of test cases relies heavily on the accuracy of reachability analysis.

The transverse goal is to push away the limitations on the use of formal verification techniques, to ease their applications, and to let them scale-up.

1. For properties that can be checked by reachability analysis we have proposed models based on regular languages and rational transductions. We have completed them by designing algorithms for verifying a refinement relation between two models $\mathcal{S}$ and $\mathcal{T}$ [67]. This refinement relation when satisfied preserves the safety properties and therefore allows them to be inherited. We shall investigate this approach with other representations.
2. In order to generate boundary-value functional test cases, we abstract models as constrained states. These constraints are solved by a customized solver, called CLPS. The test cases are derived in two steps [6]:
    1. partitioning of the formal model and extraction of boundary values,
    2. reachability graph exploration from constrained states in order to reach boundary values and generate state sequences (traces) as test cases with the oracle.

    After the generation phase, a concretization is used to produce the test drivers [7]. Furthermore, the kernel of the engine allows one to perform specification animations in order to validate the model [73].

3. For the safety of infinite state systems we have designed automated deduction tools based on term rewriting (*SPIKE*, *daTac*, *haRVey*) and an extensible and modular platform for detecting flaws and potential attacks on security protocols (*AVISPA*). The tools have been built on the modeling of systems by terms and rewrite rules. Our work with other models based on regular languages of words or trees and of transducers should complement these term rewriting models.

In order to address this challenge, we rely on complementary skills within the project. We believe that each of the three techniques will benefit from concepts and algorithms designed for the two others.

2.4. Highlights

1. In automated deduction we have contributed to a new scheme for integrating decision procedures for satisfiability problems in first-order theories and model checking in temporal logic.
2. We have started to apply successfully our verification techniques to two new application fields, namely web services and microrobotics.
3. An ERC Starting Independent Researcher Grant submitted by Véronique Cortier has been among the very few to be selected for the second phase.
4. The AVISPA Project has been nominated for Descartes Prize in 2007.

3. Scientific Foundations

3.1. Introduction

Our main goal is to design techniques and to develop tools for the verification of (safety-critical) systems, such as programs or protocols. To this end, we develop a combination of techniques based on automated deduction for program verification, constraint resolution for test generation, and reachability analysis for the verification of infinite state systems.

3.2. Automated deduction

The main goal is to prove the validity of assertions obtained from program analysis. To this end, we develop techniques and automated deduction systems based on rewriting and constraint solving. The verification of recursive data structures relies on inductive reasoning or the manipulation of equations and it also exploits some form of reasoning modulo properties of selected operators (such as associativity and/or commutativity).

Rewriting, which allows us to simplify expressions and formulae, is a key ingredient for the effectiveness of many state-of-the-art automated reasoning systems. Furthermore, a well-founded rewriting relation can be also exploited to implement reasoning by induction. This observation forms the basis of our approach to inductive reasoning, with high degree of automation and the possibility to refute false conjectures.

The constraints are the key ingredient to postpone the activity of solving complex symbolic problems until it is really necessary. They also allow us to increase the expressivity of the specification language and to refine theorem-proving strategies. As an example of this, the handling of constraints for unification problems or for the orientation of equalities in the presence of interpreted operators (e.g., commutativity and/or associativity function symbols) will possibly yield shorter automated proofs.

Finally, decision procedures are being considered as a key ingredient for the successful application of automated reasoning systems to verification problems. A decision procedure is an algorithm capable of efficiently deciding whether formulae from certain theories (such as Presburger arithmetic, lists, arrays, and their combination) are valid or not. We develop techniques to build and combine decision procedures for the domains which are relevant to verification problems. We also perform experimental evaluation of the proposed techniques by combining propositional reasoning (implemented by means of Boolean solvers – Binary Decision Diagrams or SAT solvers) and decision procedures, and their extensions to semi-decision procedures for handling larger (possibly undecidable) fragments of first-order logic.

We investigate techniques to incorporate the use of decision procedures in the model-checking of infinite state systems. The state of such systems is described by the models of theories specifying data types (such as integers or arrays) and their behavior is identified by (possibly infinite) sequences of these models which share the interpretation of the symbols interpreted in the theories (e.g., the addition over the integers). In this context, checking if a system satisfies a certain property may be reduced to checking the satisfiability of a formula in the theory obtained as the combination of the theories describing the sequence of states in the computation. To solve this problem, it is crucial to develop new combination methods for non-disjoint unions of theories.

3.3. Synthesizing and solving set constraints

Applying constraint logic programming technology in the validation and verification area is currently an active way of research. It usually requires the design of specific solvers to deal with the description language's vocabulary. We are interested in using a solver for set constraints based on the CLPS core [2], to evaluate set-oriented formal specifications. By evaluation, we mean the encoding of the formal model into a constraint system, and the ability for the solver to verify the invariant on the current constraint graph, to propagate preconditions or guards, and to apply the substitution calculus on this graph. The constraint solver is used for animating specifications and automatically generating abstract test cases.

3.4. Rewriting-based safety checking

Invariant checking and strenghtening is the dual of reachability analysis, and can thus be used for verifying safety properties of infinite-state systems. In fact, many infinite-state systems are just parameterized systems which become finite state systems when parameters are instantiated. Then, the challenge is to automatically discharge the maximal number of proof obligations coming from the decomposition of the invariance conditions. For parameterized systems, we develop a deductive approach where states are defined by first order formulae with equality, and proof obligations are checked by the automatic theorem prover *haRVey*. Thanks to this tool, we study the applicability of the superposition calculus (a modern version of resolution with a built-in treatment of the equality predicate and powerful techniques for reducing the search space) for deciding conditions arising from program verification.

4. Application Domains

4.1. Verification of security protocols

Security protocols such as SET, TLS and Kerberos, are designed for establishing the confidence of electronic transactions. They rely on cryptographic primitives, the purpose of which is to ensure integrity of data, authentication or anonymity of participants, confidentiality of transactions, etc.

Experience has shown that the design of those protocols is often erroneous, even when assuming that cryptographic primitives are perfect, i.e., that an encoded message cannot be decrypted without the appropriate key. An intruder can intercept, analyze and modify the exchanged messages with very few computations and therefore, for example, generate important economic damage.

Analyzing cryptographic protocols is complex because the set of configurations to consider is very large, and can even be *infinite*: one has to consider any number of sessions, any size of messages, sessions interleaving, algebraic properties of encryption or data structures.

Our objective is to automatize as much as possible the analysis of protocols starting from their specification. This consists in designing a tool easy to use, permitting to specify a large number of protocols thanks to a standard high-level language, and permitting either to look for flaws in a given protocol or to check whether it satisfies a given property. Such a tool is essential for verifying existing protocols, but also for helping in designing new ones. For our tool to be easy to use, it has to provide a graphical interface allowing a user to do only click-button.

Our tools for verifying security protocols are available as components of the AVISPA platform. As an extension of the AVISPA specification language, we are working on a new environment called *CASRUL* for handling more general protocols like e-business protocols for example.

4.2. Automated boundary testing from formal specifications

In [7], we have presented a new approach for test generation from set-oriented formal specifications: the BZ-TT method. This method is based on Constraint Logic Programming (CLP) techniques. The goal is to test every operation of the system at every boundary state using all input boundary values of that operation. It has been validated in several industry case studies for smart card OS and application validation (GSM 11-11 standard [68] and Java Card Virtual Machine Transaction mechanism [72]) and for embedded automotive software (an automobile wind-screen wiper controller).

This test generation method can be summed up as follows: from the formal model, the system computes boundary values to create boundary states; test cases are generated by traversal of the state space with a preamble part (sequences of operations from the initial state to a boundary state), a body part (critical invocations), an identification part (observation and Oracle state computation) and a post-amble part (return path to initial or boundary state). Then, an executable test script file is generated using a test pattern and a table of correspondence between abstract operations (from the model) and concrete ones. This approach differs in several main points from the work of Dick, Faivre *et al*: first, using boundary goals as test objectives avoids the complete construction of the reachability graph; second, this process is fully automated and the test engineer could just drive it at the boundary value computation level or for the path computation.

The BZ-TT method is fully supported by the BZ-Testing-Tools tool-set. This environment is a set of tools dedicated to animation and test cases generation from B, Z or State-Chart formal specifications. It is based on the CLPS constraint solver, able to simulate the execution of the specification. By execution, we mean that the solver computes a so-called constrained state by applying the pre- and post-condition of operations. A constrained state is a constraint store where state variables and also input and output variables support constraints.

One orientation of the current work is to go beyond the finiteness assumption limitations by using symbolic constraint propagation during the test generation process and to extend the result to object oriented specifications.

4.3. Program debugging and verification

Catching bugs in programs is difficult and time-consuming. The effort of debugging and proving correct even small units of code can surpass the effort of programming. Bugs inserted while “programming in the small” can have dramatic consequences for the consistency of a whole software system as shown, e.g., by viruses which can spread by exploiting buffer overflows, a bug which typically arises while coding a small portion of code. To detect this kind of errors, many verification techniques have been put forward such as static analysis and software model checking.

Recently, in the program verification community, there seems to be a growing demand for more declarative approaches in order to make the results of the analysis readily available to the end user[1]. To meet this requirement, a growing number of program verification tools integrate some form of theorem proving.

The goals of our research are twofold. First, we perform theoretical investigations of various combinations of propositional and first-order satisfiability checking in order to automate the theorem proving activity required to solve a large class of program analysis problems which can be encoded as first-order formulae. Second, we experimentally investigate how our techniques behave on real problems so to make program analysis more precise and scalable. Building tools capable of providing a good balance between precision and scalability is one of the crucial challenges to transfer theorem proving technology to the industrial domains.

[1] See, for example, the challenge at http://research.microsoft.com/specncheck/consel_challenge.htm.

4.4. Towards New Application Domains

4.4.1. Web services

Driven by rapidly changing requirements and business needs, IT systems and applications are undergoing a paradigm shift: components are replaced by services, distributed over the network, and composed and reconfigured dynamically in a demand-driven way into service-oriented architectures [2]. Exposing services in future network infrastructures means a wide range of trust and security issues need to be adressed. Solving them is extremely hard since making the service components trustworthy is not sufficient: composing services leads to new subtle and dangerous vulnerabilities due to interference between component services and policies, the shared communication layer, and application functionality. Thus, one needs validation of both the service components and their composition into secure service architectures. In this context, there is an obvious need of applying formal methods. Our project aims at applying our proof and constraint solving techniques to reason on web services. More precisely, we plan to focus on the composition problem in the presence of security policies.

4.4.2. Microrobotics

Researchers in microrobotics have recently proposed the concept of a distributed and integrated micromanipulator called *smart surface*, based on an array of smart micromodules in order to realize an automated positioning and conveying surface. Each micro-module will be composed of a micro-actuator, a micro-sensor and a control unit. The cooperation of these micromodules will allow to recognize the parts and to control micro-actuators on order to move and position accurately the parts on the smart surface.

Our objective is to elaborate new specification languages and verification methods to validate distributed smart surfaces at different levels of abstraction. We bring our experience in formal verification, more especially in regular model-checking (RMC). This paradigm has been studied on classical regular languages, on regular tuples of words and on regular trees. We have a good experience on these different domains. To our knowledge, there has been no attempt of applying this approach to two-dimensional (picture) languages as required for the application. Therefore, an interesting challenge is to determine how far we can follow the RMC paradigm on (regular) picture languages. In order to cope with the parametric aspect of the smart surface, we will also consider constraint propagation on formulas representing sets of configurations.

We collaborate with the "Laboratoire d'Automatique de Besançon" on verifying and validating an adaptative *microfactory* model they have developed. We have defined a complete information model of multi-cells microfactories in UML. This model is used as the communication basis between the robotic and computing researchers. It includes the structure of the physical components of the microfactory - cells and transports functions - and the logical components - information gathering and exchange [44]. The next step will be to provide properties and a dynamic model of microfactories.

5. Software

5.1. Protocols verification tools

Keywords: *Cryptography*, *Security Protocols*, *Verification*.

Participants: Laurent Vigneron, Pierre-Cyrille Héam, Heinrich Hördegen, Olga Kouchnarenko, Michaël Rusinowitch, Mathieu Turuani.

[2] see e.g. http://osoa.org/display/Main/Service+Component+Architecture+Home

5.1.1. AVISPA

Cassis has been one of the 4 partners involved in the European project AVISPA, which has resulted in the distribution of a tool for automated verification of security protocols, named *AVISPA* Tool. It is freely available on the web [3] and supported. The *AVISPA* Tool significantly extends its predecessor's scope, effectiveness, and performance, by (i) providing a modular and expressive formal language for specifying security protocols and properties, and (ii) integrating 4 back-ends that implement automatic analysis techniques ranging from *protocol falsification* (by finding an attack on the input protocol) to *abstraction-based verification* methods for both finite and infinite numbers of sessions.

In 2007, we have extended the AVISPA Tool for handling non-repudiation protocols and also for verifying the computational soundness of protocols. The first extension has been done by extending the HLPSL2IF translator for handling intruder knowledge thanks to a new predicate, *aknows*. The second extension has been done by adding a new module, permitting to transfer properties, proven in a formal model, to a computational model.

5.1.2. CL-AtSe

We develop *CL-AtSe*, a Constraint Logic based Attack Searcher for cryptographic protocols. The *CL-AtSe* approach to verification consists in a symbolic state exploration of the protocol execution, for a bounded number of sessions. This necessary restriction (for decidability, see [79]) allows *CL-AtSe* to be correct and complete, i.e., any attack found by *CL-AtSe* is a valid attack, and if no attack is found, then the protocol is secure for the given number of sessions. Each protocol step is represented by a constraint on the protocol state. These constraints are checked lazily for satisfiability, where satisfiability means reachability of the protocol state. *CL-AtSe* now includes a proper handling of sets (operations and tests), choice points, specification of any attack states through a language for expressing fairness, non-abuse freeness, etc..., advanced protocol simplifications and optimizations to reduce the problem complexity, and protocol analysis modulo the algebraic properties of cryptographic operators. In particular, *CL-AtSe* is now able to analyze protocols modulo the properties of XOR (exclusive or) or Exp (modular exponentiation). This has required to implement an optimized version of the combination algorithm of Baader & Schulz [66] for solving unification problems in disjoint unions of arbitrary theories.

In particular, *CL-AtSe* has been successfully used by Cassis members to analyse France Telecom R&D, Siemens AG, IETF, or Gemalto protocols in funded projects. It is also employed by external users, e.g., from the AVISPA's community. Moreover, *CL-AtSe* achieves very good analysis times, comparable and sometimes better than state-of-the art tools in the domain like OFMC (see [82] for tool details and precise benchmarks).

5.1.3. TA4SP

We have developed TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols), an automata based tool dedicated to the validation of security protocols for an unbounded number of sessions. This tool provides automatic computations of over and under approximations of the knowledge accessible by an intruder. This knowledge is encoded as a regular tree language and protocol steps and intruder abilities are encoded as a term rewriting system. Completions and tree automata computations are performed by Timbuk, a tool developed by project-team LANDE. When given a reachability problem such as secrecy, TA4SP reports that (1) the protocol is safe if it manages to compute an over-approximation of intruder's knowledge that does not contain a secret term or (2) the protocol is unsafe in the rewrite model if it manages to compute an underapproximation of intruder's knowledge containing a secret term or (3) I don't know otherwise. TA4SP has verified 28 industrial protocols and case (3) occurred only once, for Kaochow protocol version 2.

To efficiently handle protocols using operators with algebraic properties, TA4SP has been improved: a new quadratic completion algorithm has been implemented. Thanks to these improvements new experimental results have been obtained, for example for the Encrypted Key Exchange protocol (EKE2) using the exponential operator.

[3] http://www.avispa-project.org

As far as we know, two teams – the project-team LANDE at IRISA and the National Institute of Advanced Industrial Science and Technology in Japan – are working on the verification of security protocols using tree automata approximations. Both use tree automata dedicated tools, respectively Timbuk and Ceta-ACTAS, that can be freely downloaded on the web. However, these tools are not connected to any high level protocol specification language, and over-approximations are not fully automatically computed.

5.2. Testing tools

Keywords: *Animation of Specifications*, *CLP*, *Formal Specification*, *Test generation*.

Participants: Fabrice Bouquet, Frédéric Dadeau, Bruno Legeard.

The Testing Tools is a tool-set for animation and test generation from B, JML, Z and State-chart specifications. It consists of two components:

- **BZ-Testing-Tools** – BZ-TT – is a tool-set for animation and test generation from B, Z and State-chart specifications. BZ-TT provides several testing strategies (partitioxn analysis, cause-effect testing, boundary-value testing and domain testing), and several test model coverage criteria (multiple condition coverage, boundary coverage and transition coverage).
- **JML-Testing-Tools**[4] – JML-TT – is a framework for the symbolic animation of formal models written using JML annotations [80] embedded within Java programs. JML-TT provides a simple and efficient way to semi-automatically validate a JML specification and to check model properties such as class invariant or history constraints during the animation. This tool is used in the ACI GECCOO project[5].

We develop a third tool **Test-For-Testing-Tools** to valid the tests. The tool takes as input a code program and a test suite (realized by several approaches such as BZ-TT/random/properties driven tests). The system performs a mutation of the code program. We observe how many mutants are killed with each test suite.

5.3. Automated deduction tools: haRVey

Keywords: *Automated Deduction*, *Boolean Reasoning*, *Equational Reasoning*, *Satisfiability*, *Saturation Theorem Proving*.

Participants: Jean-François Couchot, Alain Giorgetti, Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran.

haRVey[6] is a theorem prover for first-order logic with equality [78]. It works by refutation and checks whether a first-order formula is a logical consequence of a first-order theory T, axiomatized by a finite set of formulae. Recently, the capability of reasoning in the combination of T and the theory of linear arithmetic over integers has been added. The main feature of *haRVey* is its capability of behaving as a decision procedure for the problem of checking the validity of certain classes of quantifier-free formulae modulo some theories of relevance in verification such as lists, arrays, and their combinations. The system features a combination of Boolean reasoning (supplied by a BDD or a SAT solver) to efficiently handle the boolean structure of formulae and a (generalization of the) Nelson-Oppen combination method between superposition theorem proving to flexibly reason in T and an implementation of Fourier-Motzkin method for linear arithmetic. The version of *haRVey* integrating a SAT solver has been designed and implemented by P. Fontaine (MOSEL project). *haRVey* has been especially designed to be integrated in larger verification systems. It is integrated in Barvey[7], a tool to check the consistency of B specifications. It takes a B abstract machine as input, generates proof obligations encoding the fact that the invariant is inductive, and translates them into a validity problem that *haRVey* can discharge. The tool *Why* developed by J.-C. Filliâtre (LRI, Université Paris Sud, Orsay) can generate proof obligations for *haRVey* to check the correctness of ML or C programs.

[4] http://lifc.univ-fcomte.fr/~jmltt
[5] http://geccoo.lri.fr
[6] http://www.loria.fr/equipes/cassis/softwares/haRVey/
[7] http://lifc.univ-fcomte.fr/~couchot/soft/barvey/

We are developing a JML[8] annotation generator, called JAG[9], for verifying temporal properties on Java classes. JAG consists of many translators that transform dynamic properties into standard JML annotations that ensure the satisfaction of these properties. Historically, the first input language of JAG was JTPL (Java Temporal Pattern Language), an adaptation to Java of a fragment of LTL (Linear Temporal Logic), that can deal with exceptional termination of methods and can express both safety and liveness properties.

5.4. Others tools

Most of the software tools described in previous sections are using tools that we have developed in the past: BZ-TT uses the set constraints solver CLPS; the first version of *CASRUL* was using the theorem prover *daTac*; and *SPIKE*, our induction-based theorem prover, is used in the system VOTE in collaboration with the ECOO project.

6. New Results

6.1. Automated deduction

Keywords: *Consistency*, *Decision Procedure*, *Proof*, *Satisfiability*, *Tree Automata*.

6.1.1. Decision procedures and their extensions

Participants: Enrica Nicolini, Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran, Daniele Zucchelli.

We develop general techniques which allow us to re-use available tools in order to build a new generation of satisfiability solvers offering a good trade-off between expressiveness, flexibility, and scalability. Our original approach is based on the careful integration of rewriting techniques to design satisfiability procedures for a wide range of theories formalizing data structures, together with combination techniques to build satisfiability procedures for unions of theories in a modular way.

Duc-Khanh Tran has defended his thesis [11]. The first contribution of the thesis is a rational reconstruction of the combination methods proposed by Nelson-Oppen, Shostak and others in an uniform framework. This is the starting point for further investigations. We then introduce the concept of extended canonizer and derive a modularity result for a new class of theories. This is in contrast with the lack of modularity of the class of theories considered by the Shostak method. The second contribution concerns the problem of combining rewriting-based satisfiability procedures using the Nelson-Oppen method. We use meta-saturation to develop automatic proof techniques to check important requirements for the combinability of such procedures. When meta-saturation halts for a theory, its output allows us to reason about the combinability of a rewriting-based satisfiability procedure for this theory [52]. The third contribution of this thesis is about the integration of decision procedures into SMT solvers. We consider the problem of augmenting decision procedures with the capability of computing conflict sets without degrading performances, as well as the problem of modularly constructing conflict sets for a combined theory. In this respect, we extend the Nelson-Oppen combination method to modularly build conflict sets for disjoint unions of theories. We also study how the computed conflict sets relate to an appropriate notion of minimality [54].

Program analysis and verification require decision procedures to reason on theories of data structures. In [14], we have shown the termination of a rewrite-based first-order engine on the theories of records, integer offsets, integer offsets modulo and lists. We have also given a modularity theorem stating sufficient conditions for termination on a combinations of theories, given termination on each. Finally, we have introduced several sets of benchmarks on these theories and their combinations, including both parametric synthetic benchmarks to test scalability, and real-world problems to test performances on huge sets of literals. We have compared the rewrite-based theorem prover E with the validity checkers CVC and CVC Lite. Contrary to the folklore that a general-purpose prover cannot compete with reasoners with built-in theories, the experiments are overall favorable to the theorem prover, showing that not only the rewriting approach is elegant and conceptually simple, but has important practical implications.

[8] Java Modeling Language
[9] http://jag.univ-fcomte.fr

Another important data structure for program verification is that of arrays. In [20], we have investigated extensions of the theory of arrays and designed modular decision procedures by adapting combination methods and instantiation strategies. The key idea is to add new symbols to the theory by carefully using (universal) quantifiers in their definitions and then showing that instantiating finitely many times the universally quantified variables is sufficient for completeness. Combination techniques are used to handle the theory of arrays and Presburger arithmetic over the integers. We have also explained how our techniques can be soundly integrated in state-of-the-art Satisfiability Modulo Theories solvers.

Usually, verification problems require to reason modulo a combination of theories. The concept of extended canonizers has been introduced in [81] to (i) solve the lack of modularity of Shostak combination schema and to (ii) naturally re-use rewriting-based decision procedure in combination methods. While (i) is satisfactorily discussed in [81], little is known about (ii). In [55], we have investigated the problem of efficiently implementing extended canonizers for theories of interest in verification (in particular, those of uninterpreted function symbols or lists) by adapting and combining work on rewriting-based decision procedures and SER graphs, a graph-based method defined for abstract congruence closure. Based on graphs our approach addresses implementation issues that were lacking in previous rewriting-based decision procedure approaches and which are important to argue the viability of extended canonizers.

6.1.2. Decision Procedures and Model-checking of Infinite State Systems

Participants: Enrica Nicolini, Silvio Ranise, Daniele Zucchelli.

Manna and Pnueli have extensively shown how a mixture of first-order logic (FOL) and discrete Linear time Temporal Logic (LTL) is sufficient to precisely state verification problems for the vast class of reactive systems. Theories in FOL model the (possibly infinite) data structures used by a reactive system while LTL specifies its (dynamic) behavior. The combination of LTL and FOL allows us to specify infinite state systems and the subtle ways in which their data flow influences the control flow. Indeed, the capability of automatically solving satisfiability and model-checking problems is of paramount importance to support the automation of verification techniques using this framework. In collaboration with S. Ghilardi (U. Milan), we have derived undecidability and decidability results [46], [45] for both the satisfiability of (quantifier-free) formulae and the model-checking of safety properties by lifting combination methods for (non-disjoint) theories in FOL. The proofs of our decidability results suggest how decision procedures for the constraint satisfiability problem of theories in FOL and algorithms for checking the satisfiability of propositional LTL formulae can be integrated. This paves the way to employ efficient Satisfiability Modulo Theories solvers in the model-checking of infinite state systems, as previous proposals have suggested their use for bounded model-checking.

6.1.3. Tree Automata Extensions

Participants: Michaël Rusinowitch, Laurent Vigneron.

We have considered classes of tree automata combining automata with equality test and automata modulo equational theories with F. Jacquemard (SECSI project) [21]. These tree automata are obtained by extending their standard Horn clause representations with equational conditions and rewrite systems. We show in particular that a generalized membership problem (extending the emptiness problem) is decidable by proving that the saturation of tree automata presentations with suitable paramodulation strategies terminates. Alternatively our results can be viewed as new decidable classes of first-order formula. These tree automata classes can be applied to the reachability problem for a fragment of pi-calculus that can encode protocol verification problems.

6.1.4. Verification of convergence in distributed groupware systems

Participants: Abdessamad Imine, Michaël Rusinowitch.

We are interested in the formal development of data synchronization algorithms with a proof-assistant and based on the Operational Transformation (OT) approach. For linear data structure (such as a text or an ordered XML tree) [50], we have developed a prototype environment for the collaborative edition of Wiki documents, whose concurrency control is scalable and decentralized. We are also working on ensuring data convergence in control version systems. As a case study, we have considered a text document versioning system for which we have designed a decentralized system allowing a user to compute the difference/merge operations of an arbitrary number of versions. Moreover, we are interested in differencing XML documents considered as ordered trees.

6.1.5. Automated deduction for combinatorial problems

Participant: Serge Burckel.

Serge Burckel joined the CASSIS group for the next two years. He is working in particular on applying rewriting techniques and automated deduction to knot theory. In [63], he proposes for every n, linear time reductions of the word and conjugacy problems on the braid groups B_n to the corresponding problems on the braid monoids B_n^+ and moreover only using positive words representations.

In [62], he investigates the computational complexity of three natural problems in directed acyclic graphs. He proves their NP Completeness and studies their restrictions to linear orders.

In order to construct automatic proofs, he has defined a quite expressive problem called "certified 2-SAT" which is a natural extension of the polynomial time 2-SAT problem. However, this new problem is NP-Complete. He considers also methods for simplifying a formula in DNF by deleting some literals in it. These algorithms are used for counting the number of models of a formula.

6.2. Security protocol verification

Keywords: *Exclusive-Or*, *Exponentiation*, *Protocol*, *Security*, *Verification*.

Cryptographic protocols are successfully analyzed using formal methods and many techniques have appeared in the litterature [15]. However, formal approaches usually consider the encryption schemes as black boxes and assume that an adversary cannot learn anything from an encrypted message except if he has the key. Such an assumption is too strong in general since some attacks exploit in a clever way the interaction between protocol rules and properties of cryptographic operators.

In the context of security protocol verification, two PhD theses have been defended this year. Heinrich Hoerdegen [10] has studied the link between several symbolic models and the encoding of the cryptographic primitives used in practice to translate from one model to another. He has also proposed a first setting for recursive protocols. Eugen Zalinescu [12] has obtained four main contributions in decidability and transfer results: the treatment of more involved cryptographic primitives like CBC encryption, blind signatures; a link between simple and strong secrecy [18]; the decidability of the existence of key cycles [39] and a methodology for designing secure protocols [31].

6.2.1. Extension of the Dolev-Yao model

Participants: Véronique Cortier, Michaël Rusinowitch, Mathieu Turuani.

Some attacks exploit in a clever way the interaction between protocol rules and algebraic properties of cryptographic operators. In [75], we provide a list of such properties and attacks as well as existing formal approaches for analyzing cryptographic protocols under algebraic properties.

Unbounded number of sessions. We have proposed a new class of security protocols using XOR, for which secrecy after an unbounded number of sessions is decidable [38]. The new class is important as it contains examples of key-management APIs of Hardware Security Modules, such as the IBM 4758 CCA API, which lie outside the classes for which secrecy has previously been shown to be decidable. We have further investigated this class of applications in [35] where we model *key conjuring*, the process by which an attacker obtains an unknown, encrypted key by repeatedly calling a cryptographic API function with random values in place of keys. We propose a formalism for detecting computationally feasible key conjuring operations, incorporated

into a Dolev-Yao style model of the security API. We show that security in the presence of key conjuring operations is decidable for a particular class of APIs, which includes the key management API ofIBM's Common Cryptographic Architecture (CCA).

General equational theories. We have derived decision procedures for symbolic analysis of protocols that apply to several algebraic operators associated with general classes of intruder theories.

Focusing on ground deducibility and static equivalence (checking whether two sequences of messages are indistinguishable to an attacker), we have shown [23] that decidability results can be easily combined for any disjoint equational theories: if the deducibility and indistinguishability relations are decidable for two disjoint theories, they are also decidable for their union. As an application, new decidability results can be obtained using this combination theorem. In [33], [34] we propose a general setting for solving deducibility and indistinguishability for an important class (called monoidal) of these theories. Our setting relies on the correspondence between a monoidal theory E and a semiring $\mathcal{S}_{\mathsf{E}}$ which allows us to give an algebraic characterization of the deducibility and indistinguishability problems. As a consequence we recover easily existing decidability results and obtain several new ones.

In cryptographic protocols analysis, a *treacherous* set of terms is one from which an intruder can get access to what was intended to be secret, by adding on to the top of a sequence of elements of this set, a *cap* formed of symbols legally part of his/her knowledge. In [13], we give sufficient conditions on the rewrite system modeling the intruder's abilities, such as using encryption and decryption functions, to ensure that it is decidable if such caps exist. The following classes of intruder systems are studied: linear, dwindling, Δ-strong, and optimally reducing; and depending on the class considered, the cap problem ("find a cap for a given set of terms") is shown respectively to be in P, NP-complete, decidable, and undecidable.

6.2.2. Soundness of the Dolev-Yao model

Participants: Véronique Cortier, Heinrich Hordegen, Mathieu Turuani, Eugen Zalinescu.

All the previous results rely on symbolic models of protocol executions in which cryptographic primitives are abstracted by symbolic expressions. This approach enables significantly simple and often automated proofs. However, the guarantees that it offers have been quite unclear compared to cryptographic models that consider issues of complexity and probability. Cryptographic models capture a strong notion of security, guaranteed against all probabilistic polynomial-time attacks.

We have shown in recent years that it is possible to obtain the best of both cryptographic and formal worlds in the case of public encryption: fully automated proofs and strong, clear security guarantees. Specifically, for the case of protocols that use signatures and asymmetric encryption, we have established that symbolic integrity and secrecy proofs are sound with respect to the computational model.

These soundness results require to explicitly represent the dependency of ciphertexts on randomness as labels. We have shown in [10] that for a large class of security properties (that includes rather standard formulations for secrecy and authenticity properties), security of protocols in the simpler model implies security in the label-based model. Based on these results, we have implemented an AVISPA module for verifying security properties in a standard cryptographic model.

6.2.3. Designing secure protocols

Participants: Véronique Cortier, Stéphanie Delaune, Eugen Zalinescu.

We have proposed in [31] a general transformation that maps a cryptographic protocol that is secure in an extremely weak sense (essentially in a model where no adversary is present) into a protocol that is secure against a fully active adversary which interacts with an unbounded number of protocol sessions, and has absolute control over the network. The transformation works for arbitrary protocols with any number of participants, written with usual cryptographic primitives. Our transformation provably preserves a large class of security properties that contains secrecy and authenticity.

An important byproduct contribution is a modular protocol development paradigm where designers focus their effort on an extremely simple execution setting – security in more complex settings being ensured by our generic transformation. Conceptually, the transformation is very simple, and has a clean, well motivated design. Each message is tied to the session for which it is intended via digital signatures and on-the-fly generated session identifiers, and prevents replay attacks by encrypting the messages under the recipient's public key.

Even when a protocol has been proved secure, there is absolutely no guarantee if the protocol is executed in an environment where other protocols, possibly sharing some common identities and keys like public keys or long-term symmetric keys, are executed. In [32], we show that security of protocols can be easily composed. More precisely, we show that whenever a protocol is secure, it remains secure even in an environment where arbitrary protocols are executed, provided each encryption contains some tag identifying each protocol, like e.g. the name of the protocol.

6.2.4. Security properties and advanced class of protocols

Participants: Véronique Cortier, Najah Chridi, Michaël Rusinowitch, Laurent Vigneron, Eugen Zalinescu.

Most previous results focus on secrecy and authentication for simple protocols like the ones from Clark & Jacob library. We explore several directions to cover more complex protocols and security properties.

Security Properties.

Non-repudiation protocols have an important role in many areas where secured transactions with proofs of participation are necessary. Formal methods are clever and without error, therefore using them for verifying such protocols is crucial. In this purpose, in collaboration with F. Klay (France Telecom R&D) and J. Santiago (UFRN, Natal) [61], we have shown how to partially represent non-repudiation as a combination of authentications on the Fair Zhou-Gollmann protocol. Because of the limits of this method, we have defined a new one based on the handling of the knowledge of protocol participants. This method is very general and is of natural use, as it consists in adding simple annotations, like for authentication problems. The method is very easy to implement in tools able to handle participants knowledge. We have implemented it in the AVISPA Tool and analyzed the optimistic Cederquist-Corin-Dashti protocol, discovering two unknown attacks [56]. This extension of the AVISPA Tool for handling non-repudiation opens a highway to the specification of many other properties, without any more change in the tool itself.

Some cryptographic tasks, such as contract signing and other related tasks, need to ensure complex, branching time security properties. When defining such properties one needs to deal with subtle problems regarding the scheduling of non-deterministic decisions, the delivery of messages sent on resilient (non-adversarially controlled) channels, fair executions (executions where no party, both honest and dishonest, is unreasonably precluded to perform its actions), and defining strategies of adversaries against all possible non-deterministic choices of parties and arbitrary delivery of messages via resilient channels. In [36], [37] we develop a cryptographic model that deals with all of the above problems. Based on this model and a new notion of fair scheduling, we provide a definition of a prominent branching time property of contract signing protocols, namely balance, and give the first cryptographic proof that the Asokan-Shoup-Waidner two-party contract signing protocol is balanced.

Two styles of definitions are usually considered to express that a security protocol preserves the confidentiality of a data s. Reachability-based secrecy means that s should never be disclosed while equivalence-based secrecy states that two executions of a protocol with distinct instances for s should be indistinguishable to an attacker. In [18], we have initiated a systematic investigation of the situations where syntactic secrecy entails strong secrecy. We have shown that in the passive case, reachability-based secrecy actually implies equivalence-based secrecy for digital signatures, symmetric and asymmetric encryption provided that the primitives are probabilistic. For active adversaries, we provide sufficient (and rather tight) conditions on the protocol for this implication to hold.

Group Protocols. Emerging applications require secure group communications involving hierarchical architecture protocols. Designing such secure hierarchical protocols is not straightforward, and their verification becomes a major issue in order to avoid any possible security attack and vulnerability. Several attempts have been made to deal with formal verification of group protocols, but to our knowledge, none of them did address the security of hierarchical ones. In [16], in collaboration with the MADYNES project, we have presented the specific challenges and security issues of hierarchical secure group communications, and the work that we did for their verification. We have also shown how the AtSe back-end of the AVISPA tool was used to verify one of these protocols.

6.2.5. Intruder knowledge approximation

Participants: P.-C. Héam, O. Kouchnarenko.

When the number of sessions is unbounded, the security problem of cryptographic protocols is undecidable. Hence, we have proposed automated computations of over and under-approximations of the intruder knowledge using tree automata techniques [69]. These approximation techniques are implemented in TA4SP [71], one of the tools of the AVISPA platform.

In [70], we have shown how to semi-decide whether a security protocol using algebraic properties of cryptographic primitives is safe. Our current work improves the over-approximation-based algorithms in [70] by providing a new quadratic completion algorithm to efficiently handle algebraic properties.

Recently, we have investigated the dual - insecurity - problem: in [24] we explain how to semi-decide whether a protocol using cryptographic primitive algebraic properties is unsafe. We have shown that under some constraints – often satisfied in practice – on the term rewriting systems modeling protocols it is possible to detect attacks in the rewriting model. The proposed approach for detecting attacks is currently supported by the tool TA4SP successfully applied for analysing the NSPK-xor protocol and the Diffie-Hellman protocol. However, some efforts have still to be done in order to raise a real attack from the diagnostic done in the rewriting model.

6.3. Model-based Verification and Testing

6.3.1. Regular Model-Checking on infinite Words

Keywords: *Parametric systems*, *reachability*, *regular languages*.

Participant: P.-C. Héam.

The *regular model checking* techniques use regular languages for reachability analysis: states of the system are represented by finite automata or regular expressions and actions are modelized by transducers or rewriting rules on words.

In previous work [77], [76], we have studied the language of finite words reachable from a given language by using the transitive closure of a semi-commutation relation, *i.e.* a finite union of rewriting rules

In [60], we extend this approach to infinite words : we show how to adapt the algorithms of [77] in order to handle regular ω-languages accepted by Büchi automata.

6.3.2. Model-based Testing

Participants: Fabrice Bouquet, Thibaut Brocard, Jean-François Couchot, Frédéric Dadeau, Stéphane Debricon, Alain Giorgetti, Adrien de Kermadec, Vincent Pretre.

Our model-based testing approach has been extended in three ways: i) for modelling support, ii)for security, and iii) for distributed applications.

By essence, Model-based testing is bounded to use enumerated data structures. On the other hand, formal modeling often involves parameterized data structures in order e.g. to test several implementation variants or to abstract away from irrelevant details. Hence the validation engineer has sooner or later to instantiate these parameters. Recent results in instantiation-based theorem proving show that it is often possible to guess a small instantiated formula that is equisatisfiable with the quantified one. Following this approach, we provide a formal characterization of the most general instantiation of the system. In [25] we address the problem of instantiating data structures in formal models intended to be used in a model-based testing approach.

We have introduced an original model-based testing approach that takes a UML behavioural view of the system under test and automatically generates test cases and executable test scripts according to model coverage criteria [27].

We are investigating testing techniques that exercise security properties derived from a security policy. For that we combine a functional model with a second model that formalizes a part of the security policy. Tests are computed from the security properties with the formal functional model as an oracle. This work has been performed in the project POSÉ and has been used to validate some security properties of an airport model [51].

In order to validate web service applications, we explore model based testing methodologies. The results of tests are used to compute a mark that qualify the quality of web services operations. This solution is then integrated in a validation framework based on an UDDI server. In this framework, web services are tested when they are declared to the UDDI server, and the obtained marks are supplied to customers seeking for services [53].

6.4. Verification of Web Services

6.4.1. Towards An Automatic Analysis of Web Services Security

Participants: Yannick Chevalier, Michaël Rusinowitch.

Web services send and receive messages in XML syntax with some parts hashed, encrypted or signed, according to the WS-Security standard. We have introduced [29] a model to formally describe the protocols that underly these services, their security properties and the rewriting attacks they might be subject to. Unlike other protocol models (in symbolic analysis) ours can handle non-deterministic receive/send actions and unordered sequence of XML nodes. Then to detect the attacks we have to consider the services as combining multiset operators and cryptographic ones and we have to solve specific satisfiability problems in the combined theory. By non-trivial extension of the combination techniques of [74] we obtain a decision procedure for insecurity of Web services with messages built using encryption, signature, and other cryptographic primitives. This combination technique allows one to decide insecurity in a modular way by reducing the associated constraint solving problems to problems in simpler theories.

Known protocol analysis techniques consider protocols where some piece of information expected in a protocol message is located at a fixed position. However this is too restrictive to model web-services where messages are XML semi-structured documents and where significant information (such as name, signature, ...) has to be extracted from nodes occurring at flexible positions. Therefore we have extended the Dolev Yao model by a subterm predicate [30] that allows one to express data extraction by subterm matching. This also allows one to detect so-called *rewriting attacks* that are specific to web-services.

6.4.2. Composition of Web Services

Participant: Christophe Ringeissen.

In collaboration with the ECOO project, we are working on a framework for Web services composition, including both temporal and security aspects. In our model, the composition is based on the coordination of Web services seen as a product of “conversational” automata having the capability of exchanging messages. If the coordination does not satisfy the awaited composition, we synthesize a new service, called mediator. This service aims at generating the missing messages required for the coordination so that it mimics the awaited composition [65].

The compatibility of services is a key issue for the composition problem. We are studying the compatibility problem for timed conversational automata. Our proposal relies on the inference of temporal requirements derived from local timed transitions of services. According to those inferred requirements, we can consider different forms of compatibility [48].

We are also working on applying constraint programming techniques for the composition problem. Our first contribution allows us to use a constraint modelling to instantiate a given abstract composition by selecting the most appropriate concrete Web services with respect to a query. Then, the concrete composition is built in an incremental way by propagating constraints attached to Web services. Moreover, the instantiation can be dynamically updated during the execution via a monitoring phase. This ongoing work is done in the context of the project INRIA-CONICYT CoreWeb.

6.4.3. Formalizing QoS of Web Services with Weighted Automata

Participants: Pierre-Cyrille Héam, Olga Kouchnarenko, Jérôme Voinot.

Web services are used more and more as components of distributed applications with a goal to resolve complex tasks that simple services cannot. This use of Web services is connected to the emergence of languages like WS-BPEL which allows describing the external behaviour of Web services on top of the service interfaces. The use of Web services as components of distributed applications implies the possibility to change a failing service for another which can do at least the same things as the replaced service. The composition issues are also of particular interest to Web services users. Different solutions have been proposed during the last years to check such properties, but, to our knowledge, none of them takes QoS aspects into account. In [59] we introduce underpinnings and a tool for verifying Web services substitutivity and well-formed composition while considering Web services costs such as the execution time of the different operations provided by Web services.

In this direction, the starting point of our work and the first contribution is BPEL and WSDL language extensions including several service cost notions. In [49], we have proposed to extend BPEL with a notion of service costs. In [59] we go further and consider both BPEL and WSDL specifications for being closer to the Web services reality. The main purpose of these extensions is to be able to simply specify QoS aspects of Web services. Moreover, more verification problems, e. g. strong substitutivity, well-formed composition, etc., are studied for different models, and new decision results are presented. From a theoretical point of view, we show that in a general case the substitutivity problem is undecidable, but we also point out several interesting decidable restricted cases. We also proved that the strong substitutivity problem is PSPACE-complete while it is polynomial time decidable for some interesting subclasses.

The new algorithms have been implemented in a Java based prototype that successfully works on small examples. This tool allows to automate the translation from extended BPEL/WSDL specifications into weighted automata and to automatically check Web services substitutivity and composition while considering Web services costs such as the execution time of the different operations provided by Web services. Currently, the prototype works on deterministic weighted automata (this is a very common case in practice), but in the near future we plan to extend algorithms in order to manage more general cases. The tests have been achieved on different versions of several examples like a book store example provided by Oracle[10] or the classical loan approval example.

7. Contracts and Grants with Industry

7.1. RNTL

- RNTL project DANOCOPS — *"Détection Automatique de NOn-COnformités d'un Programme vis-à-vis de ses Spécifications"*, duration: 39 months, started on 1st January 2004, ended on March 31. The goal of this project is to confront specification and program to find non-conformity. We propose

[10] http://www.oracle.com/technology/pub/articles/matjaz_bpel2.html

to use an abstract representation of specification and source program, with constraints. There are five partners, two industrials: Thales division Systèmes Aéroportés, Axlog (SS2I), and three academics: I3S/Nice, LSR/Grenoble and LIFC/Besançon. The local coordinator is F. Bouquet.

- RNTL project POSÉ — *Security policies conformance testing for embedded systems*, duration: 2 years, started in December 2005. The objective is to provide automated tools for generating tests of security policies conformance of embedded systems. There are five partners: LEIRIOS, AXALTO, SILICOMP AQ, IMAG/LSR and INRIA Lorraine. The local coordinator is F. Bouquet.

7.2. Research result transfer

The BZ-Testing-Tools technology has been transfered to LEIRIOS Technologies, at the end of 2004. The partnership between the Cassis project and the R&D LEIRIOS Department, located at the TEMIS Scientific and Industrial area at Besançon, will be continued through projects (national and international call of work) or with a new transfer protocol. According to the law of innovation, F. F. Bouquet is scientific consultant of LEIRIOS Technologies.

7.1. INTERREG

INTERREG VALID — We are working with the university of Geneve, LEIRIOS Technologies and Centre des Technologies de l'Information - État de Genève. The project concerns the test generation for the web services. The duration of the project is 18 months and it was started in July 2005.

8. Other Grants and Activities

8.1. International grants

- Project INRIA-CNPq (Brazil), DA CAPO — *Automated deduction for the verification of specifications and programs*. It is a project on the development of proof systems (like *haRVey*) for the verification of specifications and software components. The coordinators of this project are David Déharbe (UFRN Natal, Brazil) and Christophe Ringeissen. On the french side, DA CAPO also involves the PROTHEO project.
- Project INRIA-CONICYT (Chili), CoreWeb — *Constraint Reasoning for the Composition of Web Services*. The coordinators of this project are Eric Monfroy (UTFSM Valparaíso, Chili) and Michaël Rusinowitch. On the french side, CoreWeb also involves the ECOO project.
- Project INRIA-Tunisian Universities — *"Vérification et analyse de la sécurité et de la sûreté des systèmes critiques"*. The coordinators of this project are Nejib Ben Hadj-Alouane (ENSI Tunis, University Manouba) and Michaël Rusinowitch. On the french side, this project also involves the laboratory LAG (Grenoble).
- French-Tunisian project on the design and implementation of e-voting systems and of tools for verifying e-voting protocols. Duration: 2 years, started in January 2007. This is a project founded by the INRIA/DGRST, action STIC-Tunisie.

8.2. National grants

- ACI SATIN[11] —*Security Analysis for Trusted Infrastructures and Network protocols*, duration: 3 years, started on July 2004. Cassis (M. Rusinowitch) is the principal coordinator.

[11] http://lifc.univ-fcomte.fr/~heampc/SATIN

The SATIN project aims at working on formal analysis and design of secure distributed systems, by taking advantage of the recent advances in algebraic modeling techniques. Partners are: CEA-DAM, France Telecom R&D, LANDE project - IRISA, VPS team, LIFO.

- ACI Jeunes Chercheurs CRYPTO[12] —*"Lien entre la cryptanalyse et l'étude logique des protocoles cryptographiques"*, duration: 3 years, started on September 2004.

  The CRYPTO project aims at establishing a link between the formal and the computational approaches for cryptographic protocols.

- ARA SSIA FormaCrypt—*Formal proofs and probabilistic semantics in cryptography*, duration: 3 years, started in January 2006.

  The verification of cryptographic protocols is a very active research area. Most works on this topic use either the computational approach, in which messages are bitstrings, or the formal approach, in which messages are terms. The computational approach is more realistic but more difficult to automate. The FormaCrypt project aims at bringing together these orthogonal approaches in order to get the best of the two worlds. Partners are: Liens (coordinator), SECSI project - LSV, Cachan.

- ARA SSIA COPS—*Composition Of Policies and Services*, duration: 3 years, started in December 2005.

  The aim is to build technologies enabling the security analysis of web services that take into account the potential flaws at communication level, at the access policy level or at the interface between communications and access policy. Partners are: IRIT Toulouse, LIM Marseille, Microsoft R&D.

- ARA SSIA ARROWS—*Safe Pointer-Based Data Structures: A Declarative Approach to their Specification and Analysis*, duration: 3 years, started in autumn 2005.

  Programming with pointers is quite a powerful and widely used technique to build many software systems with limited resources such as embedded systems or programs requiring recursive data structures. The goal of this project is to develop new specification languages for programs manipulating pointers which are sufficiently precise to express many interesting properties and, at the same time, support automatic analyses. Partners are: CAPP-LEIBNIZ Grenoble (coordinator), LILaC-Irit Toulouse. The local coordinator is S. Ranise.

- ARA SETI RAVAJ[13] — *"Rewriting and Approximations for Java Applications Verification"*, duration: 39 months, started on January 2007. The goal of this project is to analyse MIdlets – Java programs designed for mobile devices like cell phones or PDA. In addition to classical proof tools of rewriting, we propose to use approximations of reachable terms. In order to propose a general purpose verification technique, the approximation method will be refined in different ways. In particular, we will automatically generate approximations from the property to be proved and also provide an automatic approximation refinement methodology adapted to the approximation framework.

  There are three academics partners: INRIA LANDE, INRIA PROTHEO and LIFC/Besancon; and an industrial: France Telecom R&D. The local coordinator is O. Kouchnarenko.

- QSL VALDA2—*Automated Software Verification using Automated Deduction*, duration: 2 years, started in 2005. With this action, we are working in the *Pôle de Recherche Scientifique et Technologique Intelligence Logicielle* within the theme *Qualité et sûreté des logiciels et systèmes informatiques*, funded by the *Contrat de Plan État-Région Lorraine 2000-2006*.

- QSL COWS—*Constraints for the Composition of Web Services*, duration: 2 years, started in 2006. This action is coordinated by O. Perrin (ECOO project) and L. Vigneron. It is another action of the theme *Qualité et sûreté des logiciels et systèmes informatiques*, funded by the *Contrat de Plan État-Région Lorraine 2000-2006*.

[12] http://www.loria.fr/~cortier/aci.html
[13] http://www.irisa.fr/lande/genet/RAVAJ/index.html

- SSS SeComMaNet—*Security of multicast communications in ad-hoc mobile networks*, duration: 2 years, started in 2007. This action is coordinated by L. Vigneron. This is an action of the theme *Sûreté et Sécurité des Systèmes*, funded by the Project MISN of the *Contrat de Plan État-Région Lorraine 2007-2013*.
- Competitiveness pole — *Microtechnique* and FUI[14] Project VALMI - *Validation automatique de microsystèmes embarqués de transaction électronique en billétique*. Duration : 18 months, started in November 2006. The aim of this project is to provide automated tools for generation tests of embedded system around distribution and validation of urban travel pass. There are four partners: ERG, Leirios, Parkeon and LIFC. The local coordinator is F. Bouquet.
- ANR program "Systèmes interactifs et robotique"— *Smart Surface*, coordinated by the "Laboratoire d'Automatique de Besançon, UMR CNRS 6596". This project started in July 2007 for three years. The project is a multidisciplinary collaboration between national and international research teams with large skills in microsystems and microtechnolgies with FEMTO-ST (Besançon) and LIMMS (Tokyo), in design methods and technologies with l'InESS (Strasbourg), in distributed and network based processing and formal verification with the LIFC (Besançon) and LAAS (Toulouse) and in automated control of microrobotics systems with the LAB (Besançon). The local coordinator is Alain Giorgetti.

8.3. International collaborations

- In the area of automated test generation from a formal model, we have an active collaboration with Dr Mark Utting from the Formal Method group from the University of Waikato[15]. This cooperation is supported by the France-New-Zealand scientific program.
- In the area of business applications, we are working on the soundness problem of coloured work-flow Petri nets with the Information System group of Professor K. van Hee from the Technical University of Eindhoven. This cooperation is supported by the NWO scientific program (The Netherlands).

8.4. Individual involvement

F. Bouquet: Coordinator of Tools session of B'07. PC Member of 9th Workshop on *Formal Techniques for Java-like Programs*, FTfJP 2007 a satellite workshop of ECOOP 2007. Member of the technical program committee of the first IEEE *International Conference on Software Testing, Verification and Validation(ICST 2008)*. Inria expert for the OPEES (Open Platform for Engineering of Embedded Systems) project in the workpackage *Vérification de Modèles, Génération de tests*.
V. Cortier: local coordinator of the ARA SSIA FormaCrypt (started in January 2006); coordinator of the ACI Jeunes Chercheurs CRYPTO; French coordinator of the French-Tunisian project on e-voting; PC member of *the 33rd International Colloquium on Automata, Languages and Programming, Security Track (Icalp'07, Track C), Wroclaw, Poland*, *the 5th ACM Workshop on Formal Methods in Security Engineering (FMSE'07), Alexandria, USA*, *the 5th International Workshop on Security Issues in Concurrency (SecCo 2007), Lisboa, Portugal.*, *the workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA'07), Wroclaw, Poland.*, *the Workshop on Formal and Computational Cryptography (FCC 2007), Venice, Italy.*, *the Workshop on Security and Rewriting Techniques (SecReT 2007), Paris, France*, organizer and PC member of the Workshop *La Sécurité Informatique et le Vote ElecTrOnique (VETO'07)*, Tunisia; member of the "CSE 27" of the ENS Cachan and of the INPL.
A. Giorgetti: Editorial committee member of *Techniques et Science Informatique (TSI)*.
P.-C. Héam: member of the "CSE 27" of the University of Franche-Comté.

[14] Fonds Unique Interministériel & Fonds de Compétitivité des Entreprises (FCE)
[15] http://www.cs.waikato.ac.nz/Research/fm/index.html

O. Kouchnarenko: director of the research team *Techniques Formelles et à Contraintes (TFC)* of the *Laboratoire d'informatique de Franche Comté (LIFC)*; Co-chair of the "*Formal Specification and Development in B*", B'2007, and PC member of "*Approches Formelles dans l'Assistance au Développement de Logiciels*", AFADL'07. Member of the "CSE 27" of the University of Franche-Comté, coordinator of the "Licence Informatique 2008-2012" of the LMD2 project of the University of Franche-Comté.
S. Ranise: trustee of the project CALCULEMUS (Systems for Integrated Computation and Deduction); coordinator (with Cesare Tinelli) of the Satisfiability Modulo Theories Library (SMT-LIB) initiative; PC chair of FTP 2007 and PC member of FroCoS 2007.
C. Ringeissen: PC member of FroCoS 2007 and FTP 2007.
M. Rusinowitch: member of the IFIP Working Group 1.6 (Rewriting); coordinator of the project ACI Sécurité SATIN. PC member of COLSEC, Workshop on Collaboration and Security 2007, Irvine, California, USA; CRISIS 2007, International Conference on Risks and Security of Internet Systems, Marrakech, 2-5 July 2007. Member of the "CSE 27" of Nancy 2 University and Institut National Polytechnique de Lorraine.
L. Vigneron: member of the FTP steering committee; Secretary of the IFIP Working Group 1.6 (Rewriting); Organising Committee Chair of the second International School on Rewriting, ISR'2007; Webmaster of the site *Rewriting Home Page*, of the RTA conference site, and of the web page for the IFIP Working Group 1.6.
We are involved in several lectures of the "Master Informatique" of the universities of Nancy. V. Cortier is in charge of the lecture on *Theory of the security*, S. Ranise and C. Ringeissen are in charge of the lecture on *Decision procedures and program verification*.

8.5. Visits of foreign researchers

S.P. Suresh: from CMI, Chennai visited Cassis for 2 weeks in June 2007 and has worked with E. Zalinescu on modelling security with epistemic logics.
M. Utting: from August 30 to September 7, 2007 for working on model-based testing.
H. Boucheneb: professeur agrégé, Université de Montréal, has been invited professor in CASSIS during 2 months (November/December) and has been working on infinite state verification.
In 2007/08, Dr. *Natalia Sidorova* will be an Invited Professor at the University of Franche-Comté.

9. Dissemination

9.1. Ph. D. theses

Heinrich Hoerdegen has defended his Ph. D. thesis, entitled "*Vérification des protocoles cryptographiques : Comparaison des modèles symboliques avec une application des résultats — Étude des protocoles récursifs*", supervised by V. Cortier and M. Rusinowitch, on November 29, 2007.
Duc-Khanh Tran has defended his Ph. D. thesis, entitled "*Conception de procédures de décision par combinaison et saturation*", supervised by H. Kirchner, S. Ranise and C. Ringeissen, on February 16, 2007.
Eugen Zalinescu has defended his Ph. D. thesis, entitled "*Sécurité des protocoles cryptographiques: décidabilité et résultats de transfert*", supervised by V. Cortier and M. Rusinowitch, on December 17, 2007.

9.2. Committees

F. Bouquet is referee for the theses of Patricia Mouy, May 16, 2007, CEA (Univ. of Evry), and of Jeremy Briffaut, December 17,2007, ENSI-Bourges (Univ. of Orléans).
A. Giorgetti is examiner for the thesis of Julien Groslambert, September 14, 2007.
O. Kouchnarenko is member of the ASTI committee to award the best Ph. D. dissertations of the *Fédération des Associations Fran caises des Sciences et Technologies de l'Informations*. O. Kouchnarenko is examiner/chair for the theses of Julien Groslambert, September 14, 2007, University of Franche-Comté, and of Pierre Pillot, December 4, 2007, University of Orléans.
M. Rusinowitch is referee for the thesis of Andrei Paskevich (Paris) and examiner/chair for the thesis of Fabrice Nahon (Nancy).

9.3. Seminars, workshops, and conferences

Besides conference talks mentioned in the publication list, we have participated to the following events.
A. GIORGETTI, *Model-checking par génération d'annotations*, tool demonstration at CNRS evaluation committee, Besançon, February 1st 2007.
C. RINGEISSEN, participation to the Dagstuhl seminar *Automated Deduction and Decision Procedures*, October 1-5, 2007.
M. RUSINOWITCH, Invited Talk on Some Decision Problems Related to Cryptographic Protocol Verification, Automated Deduction: Decidability, Complexity, Tractability (ADDCT'07) Affiliated with CADE-21 Bremen, Germany, 15 July, 2007.
M. RUSINOWITCH, *Tutorial on Security*, International School on Rewriting ISR 2007, Nancy, July 3rd 2007.
M. RUSINOWITCH, Keynote Talk on *Automated Verification of Cryptographic Protocols*, at the Workshop on Rigorous System Development and Analysis. Nancy, October 12, 2007.
L. VIGNERON, participation to the Dagstuhl seminar *Formal Protocol Verification Applied*, October 15-19, 2007.

10. Bibliography

Major publications by the team in recent years

[1] A. ARMANDO, S. RANISE, M. RUSINOWITCH. *A Rewriting Approach to Satisfiability Procedures*, in "Journal of Information and Computation — Special Issue on Rewriting Techniques and Applications (RTA'01)", vol. 183, n^{o} 2, June 2003, p. 140–164.

[2] F. BOUQUET, B. LEGEARD, F. PEUREUX. *CLPS-B: A Constraint Solver to Animate a B Specification*, in "International Journal of Software Tools for Technology Transfer, STTT", vol. 6, n^{o} 2, August 2004, p. 143–157.

[3] Y. CHEVALIER, L. VIGNERON. *Strategy for Verifying Security Protocols with Unbounded Message Size*, in "Journal of Automated Software Engineering", vol. 11, n^{o} 2, April 2004, p. 141–166.

[4] H. COMON-LUNDH, V. CORTIER. *Security properties: two agents are sufficient*, in "Science of Computer Programming", vol. 50, n^{o} 1-3, March 2004, p. 51–71, http://www.loria.fr/~cortier/Papiers/ComonCortierSCP03.ps.

[5] F. JACQUEMARD, M. RUSINOWITCH, L. VIGNERON. *Compiling and Verifying Security Protocols*, in "Logic for Programming and Automated Reasoning (LPAR'00), Reunion Island, France", A. VORONKOV, M. PARIGOT (editors), Lecture Notes in Computer Science, vol. 1955, Springer, 2000, p. 131–160.

[6] B. LEGEARD, F. PEUREUX. *B-Testing-Tools : génération de tests aux limites à partir de spécifications B*, in "TSI, Techniques et Sciences Informatiques, Hermès-Lavoisier", vol. 21, n^{o} 9, 2002, p. 1189–1218.

[7] B. LEGEARD, F. PEUREUX, M. UTTING. *Automated Boundary Testing from Z and B*, in "Formal Methods Europe (FME 2002)", L.-H. ERIKSSON, P. LINDSAY (editors), Lecture Notes in Computer Science, vol. 2391, Springer, 2002, p. 21–40.

[8] M. RUSINOWITCH, M. TURUANI. *Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete*, in "Theoretical Computer Science", vol. 299, April 2003, p. 451–475, http://www.loria.fr/~rusi/pub/tcsprotocol.ps.gz.

[9] C. TINELLI, C. RINGEISSEN. *Unions of Non-Disjoint Theories and Combinations of Satisfiability Procedures*, in "Theoretical Computer Science", vol. 290, n^{o} 1, 2003, p. 291–353.

Year Publications

Doctoral dissertations and Habilitation theses

[10] H. HOERDEGEN. *Vérification des protocoles cryptographiques : Comparaison des modèles symboliques avec une application des résultats — Étude des protocoles récursifs*, Thèse de Doctorat, Université Henri Poincaré, Nancy, France, Novembre 2007.

[11] D.-K. TRAN. *Conception de procédures de décision par combinaison et saturation*, Thèse de Doctorat, Université Henri Poincaré, Nancy, France, Février 2007, http://www.loria.fr/~tran/Thesis/PhD-Tran.pdf.

[12] E. ZALINESCU. *Sécurité des protocoles cryptographiques: décidabilité et résultats de transfert*, Thèse de Doctorat, Université Henri Poincaré, Nancy, France, Décembre 2007.

Articles in refereed journals and book chapters

[13] S. ANANTHARAMAN, P. NARENDRAN, M. RUSINOWITCH. *Intruders with Caps*, F. BAADER (editor), Lecture Notes in Computer Science, vol. 4533, Springer, Paris, France, June 2007, p. 20-35, http://www.loria.fr/~rusi/pub/rta07.pdf.

[14] A. ARMANDO, M. P. BONACINA, S. RANISE, S. SCHULZ. *New Results on Rewrite-based Satisfiability Procedures*, in "ACM Transaction on Computational Logic", To appear, 2007.

[15] Y. BOICHUT, P.-C. HÉAM, O. KOUCHNARENKO. *Vérifier automatiquement les protocoles de sécurité*, in "Techniques de l'ingénieur", A paraître, October 2007, p. ***–***.

[16] M. S. BOUASSIDA, N. CHRIDI, I. CHRISMENT, O. FESTOR, L. VIGNERON. *Automated Verification of a Key Management Architecture for Hierarchical Group Protocols*, in "Annals of Telecommunications", To appear, 2007.

[17] N. CHRIDI, L. VIGNERON. *24*, in "Strategy for Flaws Detection based on a Services-driven Model for Group Protocols", Future and Trends in Constraint Programming, ISTE, April 2007, p. 361-370.

[18] V. CORTIER, M. RUSINOWITCH, E. ZALINESCU. *Relating two Standard Notions of Secrecy*, in "Logical Methods in Computer Science", vol. 3, n^{o} 3, July 2007, http://www.lmcs-online.org/ojs/viewarticle.php?id=273&layout=abstract.

[19] G. CÉCÉ, P.-C. HÉAM, Y. MAINIER. *Efficiency of Automata in Semi-Commutation Verification Techniques*, in "Theoretical Informatics and Applications", Also available as Research Report 5001, INRIA, France, 2007.

[20] S. GHILARDI, E. NICOLINI, S. RANISE, D. ZUCCHELLI. *Decision Procedures for Extensions of the Theory of Arrays*, in "Annals of Mathematics and Artificial Intelligence", vol. 50, n^{o} 3-4, 2007, p. 231-254.

[21] F. JACQUEMARD, M. RUSINOWITCH, L. VIGNERON. *Tree Automata with Equality Constraints Modulo Equational Theories*, in "Journal of Logic and Algebraic Programming", Accepted for publication, 2007.

Publications in Conferences and Workshops

[22] T. ABBES, A. BOUHOULA, M. RUSINOWITCH. *A Traffic Classification Algorithm for Intrusion Detection*, in "Symposium on Frontiers in Networking with Applications (FINA-07), Workshops Proceedings, Niagara Falls, Canada", vol. 1, IEEE Computer Society, May 2007, p. 188-193.

[23] M. ARNAUD, V. CORTIER, S. DELAUNE. *Combining algorithms for deciding knowledge in security protocols*, in "Proceedings of the 6th International Symposium on Frontiers of Combining Systems (FroCoS'07), Liverpool, UK", F. WOLTER (editor), Lecture Notes in Artificial Intelligence, vol. 4720, Springer, September 2007, p. 103-117, http://www.loria.fr/~cortier/Papiers/FROCOS07.pdf.

[24] Y. BOICHUT, P.-C. HÉAM, O. KOUCHNARENKO. *Tree Automata for Detecting Attacks on Protocols with Algebraic Cryptographic Primitives*, in "INFINITY'07, Int. Ws. on Verification of Infinite-State Systems, joint to CONCUR'07, Lisboa, Portugal", The final version will be published in EN in Theoretical Computer Science, Elsevier, September 2007, p. 44-53.

[25] F. BOUQUET, J.-F. COUCHOT, F. DADEAU, A. GIORGETTI. *Instantiation of Parameterized Data Structures for Model-Based Testing*, in "B'2007, the 7th Int. B Conference, Besancon, France", Lecture Notes in Computer Science, vol. 4355, Springer, January 2007, p. 96–110.

[26] F. BOUQUET, F. DADEAU, J. GROSLAMBERT. *JML2B: Checking JML specifications with B machines*, in "7th Int. B Conference - Tool Session, Besancon, France", Lecture Notes in Computer Science, vol. 4355, Springer, January 2007, p. 285–288.

[27] F. BOUQUET, C. GRANDPIERRE, B. LEGEARD, F. PEUREUX, N. VACELET, M. UTTING. *A subset of precise UML for model-based testing*, in "A-MOST'07: Proceedings of the 3rd international workshop on Advances in Model-based Testing, London, United Kingdom", ACM Press, July 2007, p. 95-104.

[28] S. BURSUC, H. COMON-LUNDH, S. DELAUNE. *Deducibility Constraints, Equational Theory and Electronic Money*, in "Rewriting, Computation and Proof — Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of his 60th Birthday, Cachan, France", H. COMON-LUNDH, C. KIRCHNER, H. KIRCHNER (editors), Lecture Notes in Computer Science, vol. 4600, Springer, June 2007, p. 196-212, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/BCD-jpj07.ps.

[29] Y. CHEVALIER, D. LUGIEZ, M. RUSINOWITCH. *Towards an Automatic Analysis of Web Service Security*, in "Proceedings of the 6th International Symposium on Frontiers of Combining Systems (FroCoS'07), Liverpool, UK", F. WOLTER (editor), Lecture Notes in Artificial Intelligence, vol. 4720, Springer, September 2007, p. 133-147, http://www.loria.fr/~rusi/pub/frocos07.pdf.

[30] Y. CHEVALIER, D. LUGIEZ, M. RUSINOWITCH. *Verifying Cryptographic Protocols with Subterms Constraints*, in "Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan, Armenia, Proceedings, Erevan, Armenie", N. DERSHOWITZ, A. VORONKOV (editors), Lecture Notes in Artificial Intelligence, vol. 4790, Springer, October 2007, p. 181-195, http://www.loria.fr/~rusi/pub/lpar07.pdf.

[31] V. CORTIER, W. BOGDAN, E. ZALINESCU. *Synthetizing Secure Protocols*, in "Proceedings of the 12th European Symposium On Research In Computer Security (ESORICS'07), Dresden, Germany", vol. 4734, Springer, September 2007, p. 406-421, http://www.loria.fr/~cortier/Papiers/compiler.pdf.

[32] V. CORTIER, J. DELAITRE, S. DELAUNE. *Safely Composing Security Protocols*, in "Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India", V. ARVIND, S. PRASAD (editors), Lecture Notes in Computer Science, To appear, Springer, December 2007, http://www.loria.fr/~cortier/Papiers/FSTTCS07.pdf.

[33] V. CORTIER, S. DELAUNE. *Deciding Knowledge in Security Protocols for Monoidal Equational Theories*, in "Proceedings of the Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA'07), Wrocław, Poland", P. DEGANO, R. KÜSTERS, L. VIGANÒ, S. ZDANCEWIC (editors), July 2007, p. 63-80, http://www.loria.fr/~cortier/Papiers/FCS-ARSPA07.pdf.

[34] V. CORTIER, S. DELAUNE. *Deciding Knowledge in Security Protocols for Monoidal Equational Theories*, in "Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan, Armenia, Proceedings, Yerevan, Armenia", Lecture Notes in Artificial Intelligence, vol. 4790, Springer, October 2007, p. 196-210, http://www.loria.fr/~cortier/Papiers/LPAR07.pdf.

[35] V. CORTIER, S. DELAUNE, G. STEEL. *A Formal Theory of Key Conjuring*, in "Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF'07), Venice, Italy", IEEE Computer Society Press, July 2007, p. 79-93, http://www.loria.fr/~cortier/Papiers/CSF07.pdf.

[36] V. CORTIER, R. KÜSTERS, B. WARINSCHI. *A Cryptographic Model for Branching Time Security Properties – the Case of Contract Signing Protocols*, in "Proceedings of the 12th European Symposium On Research In Computer Security (ESORICS'07), Dresden, Germany", vol. 4734, Springer, September 2007, p. 422-437.

[37] V. CORTIER, R. KÜSTERS, B. WARINSCHI. *A Cryptographic Model for Branching Time Security Properties – the Case of Contract Signing Protocols*, in "3rd Workshop on Formal and Computational Cryptography (FCC 2007), Venice, Italy", July 2007.

[38] V. CORTIER, G. KEIGHREN, G. STEEL. *Automatic Analysis of the Security of XOR-based Key Management Schemes*, in "13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07), Braga, Portugal", Lecture Notes in Computer Science, vol. 4424, Springer, March 2007, p. 538-552, http://www.loria.fr/~cortier/Papiers/TACAS07.pdf.

[39] V. CORTIER, E. ZALINESCU. *Deciding key cycles for security protocols*, in "3rd Workshop on Formal and Computational Cryptography (FCC 2007), Venice, Italy", July 2007.

[40] S. DELAUNE, S. KREMER, M. D. RYAN. *Symbolic Bisimulation for the Applied Pi Calculus*, in "Proceedings of the 5th International Workshop on Security Issues in Concurrency (SecCo'07), Lisbon, Portugal", D. GORIA, C. PALAMIDESSI (editors), Electronic Notes in Theoretical Computer Science, Elsevier Science Publishers, September 2007, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DKR-secco07.pdf.

[41] S. DELAUNE, S. KREMER, M. D. RYAN. *Symbolic Bisimulation for the Applied Pi-Calculus*, in "Proceedings of the 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07), New Delhi, India", V. ARVIND, S. PRASAD (editors), Lecture Notes in Computer Science, To appear, Springer, December 2007, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DKR-fsttcs07.pdf.

[42] S. DELAUNE, H. LIN, CH. LYNCH. *Protocol Verification via Rigid/Flexible Resolution*, in "Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan, Armenia, Proceedings, Yerevan, Armenia", N. DERSHOWITZ, A. VORONKOV (editors), Lecture Notes in Artificial Intelligence, vol. 4790, Springer, October 2007, p. 242-256.

[43] S. DELAUNE, H. LIN, CH. LYNCH. *Protocol verification via rigid/flexible resolution*, in "Proceedings of the Workshop on Automated Deduction: Decidability, Complexity, Tractability (ADDCT'07), Bremen, Germany", S. GHILARDI, U. SATTLER, V. SOFRONIE-STOKKERMANS, A. TIWARI (editors), To appear, July 2007.

[44] E. DESCOURVIÈRES, S. DEBRICON, D. GENDREAU, P. LUTZ, L. PHILIPPE, F. BOUQUET. *Towards automatic control for microfactories*, in "5th Int. Conf. on Industrial Automation, Montréal, Québec, Canada", ETS, June 2007.

[45] S. GHILARDI, E. NICOLINI, S. RANISE, D. ZUCCHELLI. *Combination Methods for Satisfiability and Model-Checking of Infinite-State Systems*, in "Proceedings of the 21st Conference on Automated Deduction (CADE 2007), Bremen (Germany)", F. PFENNING (editor), Lecture Notes in Computer Science, vol. 4603, Springer, 2007, p. 362-378.

[46] S. GHILARDI, E. NICOLINI, S. RANISE, D. ZUCCHELLI. *Noetherianity and Combination Problems*, in "Proceedings of the 6th International Workshop on Frontiers of Combining Systems (FroCoS 2007), Liverpool (UK)", B. KONEV, F. WOLTER (editors), Lecture Notes in Computer Science, vol. 4720, Springer, 2007, p. 206-220.

[47] A. GIORGETTI, J. GROSLAMBERT. *Un programme annoté en vaut deux*, in "Journées francophones des langages applicatifs (JFLA'07), Aix-les-Bains, France", P.-E. MOREAU (editor), INRIA, January 2007, p. 87-101.

[48] N. GUERMOUCHE, O. PERRIN, C. RINGEISSEN. *Timed Specification For Web Services Compatibility Analysis*, in "Proc. of 3rd International Workshop on Automated Specification and Verification of Web Systems, WWV", December 2007.

[49] P.-C. HEAM, O. KOUCHNARENKO, J. VOINOT. *How to Handle QoS Aspects in Web Services Substitutivity Verification*, in "International Workshop on Information Systems & Web Services, as part of the 16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2007), Paris, France", June 2007.

[50] A. IMINE, M. RUSINOWITCH. *Applying a Theorem Prover to the Verification of Optimistic Replication Algorithms*, in "Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday, Cachan, France", H. COMON-LUNDH, C. KIRCHNER, H. KIRCHNER (editors), Lecture Notes in Computer Science, vol. 4600, Springer, June 2007, p. 196-212.

[51] R. LALEAU, Y. LEDRU, D. BERT, F. BOUQUET, M. LEMOINE, C. DUBOIS, S. VIGNES, V. VIGUIÉ DONZEAU-GOUGE. *Using Computer Science Modeling Techniques for Airport Security Certification*, in "RCIS'07, 1st Int. Conf. on Research Challenges in Information Science, Ouarzazate, Morocco", April 2007, p. 61-72.

[52] C. LYNCH, D.-K. TRAN. *Automatic Decidability and Combinability Revisisted*, in "Conference on Automated Deduction (CADE'07), Bremen, Germany", F. PFENNING (editor), Lecture Notes in Computer Science, vol. 4603, July 2007, p. 328-344.

[53] V. PRETRE, F. BOUQUET, C. LANG. *A Model-Based Validation Framework for Web Services*, in "ICSSEA'2007 - 5th International Workshop on System Testing and Validation (STV'2007), Paris, France", To appear, Fraunhofer book series, December 2007.

[54] S. RANISE, C. RINGEISSEN, D.-K. TRAN. *Combining Proof Producing Decision Procedures*, in "Frontier of Combining System (FroCos'07), Liverpool, UK", F. WOLTER (editor), Lecture Notes in Computer Science, vol. 4720, Springer, September 2007, p. 237-251.

[55] S. RANISE, C. SCHARFF. *Building Extended Canonizers by Graph-Based Deduction*, in "4th Int. Colloquium on Theoretical Aspects of Computing (ICTAC'07), Macao, SAR, China", Lecture Notes in Computer Science, vol. 4711, Springer, September 2007, p. 440-454.

[56] J. SANTIAGO, L. VIGNERON. *Optimistic Non-repudiation Protocol Analysis*, in "Proceedings of the Workshop in Information Security Theory and Practices (WISTP'2007), Smart Cards, Mobile and Ubiquitous Computing Systems, Heraklion (Greece)", D. SAUVERON, ET AL (editors), Lecture Notes in Computer Science, vol. 4462, Springer, May 2007, p. 90-101, http://www.loria.fr/~vigneron/Work/papers/SantiagoV-WISTP07.pdf.

Internal Reports

[57] M. ARNAUD, V. CORTIER, S. DELAUNE. *Combining algorithms for deciding knowledge in security protocols*, 28 pages, Research Report, n^{o} 6118, INRIA, February 2007, http://hal.inria.fr/inria-00129418.

[58] V. CORTIER, S. DELAUNE, G. STEEL. *A Formal Theory of Key Conjuring*, 38 pages, Research Report, n^{o} 6134, INRIA, February 2007, http://hal.inria.fr/inria-00129642.

[59] P.-C. HEAM, O. KOUCHNARENKO, J. VOINOT. *Towards Formalizing QoS of Web Services with Weighted Automata*, 22 pages, Research Report, n^{o} RR-6218, INRIA, June 2007, https://hal.inria.fr/inria-00154453.

[60] P.-C. HÉAM. *Transitive Closures of Semi-commutation Relations on Regular omega-Languages*, 20 pages, Research Report, n^{o} RR-6239, INRIA, June 2007, https://hal.inria.fr/inria-00158285.

[61] F. KLAY, J. SANTIAGO, L. VIGNERON. *Automatic Methods for Analyzing Non-Repudiation Protocols with an Active Intruder*, Research Report, n^{o} 6324, INRIA, October 2007, https://hal.inria.fr/inria-00179550/en/.

Miscellaneous

[62] S. BURCKEL. *Complexity of some Path Problems in DAGs and Linear Orders*, 2007, arXiv:0710.2268.

[63] S. BURCKEL. *Reduce Problems From Braid Groups To Braid Monoids*, 2007, arXiv:0709.3887.

[64] R. COURBIS. *Raffinement d'approximations pour la vérification de MIDlet*, Mémoire de Master Recherche, LIFC, Université de Franche-Comté, 2007.

[65] N. GUERMOUCHE, O. PERRIN, C. RINGEISSEN. *A Mediator Based Approach For Services Composition*, 2007, Research report.

References in notes

[66] F. BAADER, K. U. SCHULZ. *Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures*, in "Journal of Symbolic Computation", vol. 21, n^{o} 2, February 1996, p. 211–243.

[67] F. BELLEGARDE, C. DARLOT, J. JULLIAND, O. KOUCHNARENKO. *Reformulation: a Way to Combine Dynamic Properties and Refinement*, in "International Symposium Formal Methods Europe (FME 2001)", LNCS, vol. 2021, Springer-Verlag, 2001.

[68] E. BERNARD, B. LEGEARD, X. LUCK, F. PEUREUX. *Generation of Test Sequences from Formal Specifications: GSM 11-11 Standard Case-Study*, in "International Journal on Software Practice and Experience", vol. 34, n^{o} 10, 2004, p. 915–948.

[69] Y. BOICHUT. *Approximations pour la vérification automatique de protocoles de sécurité*, Thèse de Doctorat, LIFC, Université de Franche-Comté, Besançon (France), septembre 2006.

[70] Y. BOICHUT, P.-C. HÉAM, O. KOUCHNARENKO. *Handling Algebraic Properties in Automatic Analysis of Security Protocols*, in "3rd International Colloquium on Theoretical Aspects of Computing, ICTAC, Tunis, Tunisia", Lecture Notes in Computer Science, vol. 4281, November 2006, p. 153–167.

[71] Y. BOICHUT, N. KOSMATOV, L. VIGNERON. *Validation of Prouve Protocols using the Automatic Tool TA4SP*, in "Proceedings of 3rd Taiwanese-French Conference on Information Technology (TFIT), Nancy, France", March 2006, p. 467-480.

[72] F. BOUQUET, B. LEGEARD. *Reification of Executable Test Scripts in Formal Specification-Based Test Generation: The Java Card Transaction Mechanism Case Study*, in "Formal Methods, FME 2003", vol. 2805, Springer-Verlag, September 2003, p. 778–795.

[73] F. BOUQUET, B. LEGEARD, F. PEUREUX. *CLPS-B - A Constraint Solver for B*, in "International Conference on Tools and Algorithms for Construction and Analysis of Systems, TACAS2002, Grenoble, France", Lecture Notes in Computer Science, vol. 2280, Springer, April 2002, p. 188–204.

[74] Y. CHEVALIER, M. RUSINOWITCH. *Combining Intruder Theories*, in "Proc. of the Int. Coll. on Automata, Languages and Programming, ICALP, Lisbon, Portugal", Lecture Notes in Computer Science, vol. 3580, Springer, 2005, p. 639-651.

[75] V. CORTIER, S. DELAUNE, P. LAFOURCADE. *A Survey of Algebraic Properties Used in Cryptographic Protocols*, in "Journal of Computer Security", vol. 14, n^{o} 1, 2006, p. 1–43, http://www.loria.fr/~cortier/Papiers/survey.ps.

[76] G. CÉCÉ, P.-C. HÉAM, Y. MAINIER. *Clôtures transitives de semi-commutations et model-checking régulier*, in "Congrès Approches Formelles dans l'Assistance au Développement de Logiciels, AFADL'04, Besançon, France", J. JULLIAND (editor), June 2004, p. 257–268.

[77] G. CÉCÉ, P.-C. HÉAM, Y. MAINIER. *Efficiency of Automata in Semi-Commutation Verification Techniques*, in "Theoretical Informatics and Applications", Also available as Research Report 5001, INRIA, France, 2004, http://hal.inria.fr/inria-00077039.

[78] D. DÉHARBE, S. RANISE. *Light-Weight Theorem Proving for Debugging and Verifying Units of Code*, in "Proc. of the International Conference on Software Engineering and Formal Methods (SEFM03), Brisbane, Australia", IEEE Computer Society Press, September 2003, http://www.loria.fr/~ranise/pubs/sefm03.ps.gz.

[79] S. EVEN, O. GOLDREICH. *On the Security of Multi-Party Ping-Pong Protocols*, in "IEEE Symposium on Foundations of Computer Science", 1983, p. 34-39, http://citeseer.ist.psu.edu/46982.html.

[80] G. T. LEAVENS, A. L. BAKER, C. RUBY. *JML: a Java Modeling Language*, in "Formal Underpinnings of Java Workshop (at OOPSLA '98)", October 1998.

[81] S. RANISE, C. RINGEISSEN, D.-K. TRAN. *Nelson-Oppen, Shostak and the Extended Canonizer: A Family Picture with a Newborn*, in "First International Colloquium on Theoretical Aspects of Computing - ICTAC 2004, Guiyang, Chine", Lecture Notes in Computer Science, vol. 3407, Springer, September 2004, p. 372-386.

[82] M. TURUANI. *The CL-AtSe Protocol Analyser*, in "Term Rewriting and Applications - Proc. of RTA, Seattle, WA, USA", Lecture Notes in Computer Science, vol. 4098, 2006, p. 277–286.