INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Mosel

Proof-oriented development of computer-based systems

Lorraine

THEME SYM

Table of contents

1. Team

MOSEL is a project of INRIA Lorraine (since February 2004) and LORIA (UMR 7503). It is affiliated with the Université Henri Poincaré Nancy 1, the Université Nancy 2, the Institut national polytechnique de Lorraine, the CNRS, and INRIA Lorraine. One member of MOSEL is a member of the faculty of the University of Metz.

Team Leader

Dominique Méry [Professor, University Henri Poincaré Nancy 1, HdR]

Administrative Assistant

Josiane Reffort [Administrative Assistant, University Henri Poincaré Nancy 1]

INRIA Staff

Stephan Merz [Research Director, INRIA, HdR]

University Staff

Dominique Cansell [Assistant Professor, University of Metz, HdR]
Pascal Fontaine [Assistant Professor, University Nancy 2]
Jacques Jaray [Professor, Institut National Polytechnique de Lorraine and École des Mines de Nancy, HdR]
Leonor Prensa Nieto [Assistant Professor, Institut National Polytechnique de Lorraine and École Nationale Supérieure d'Electricité et de Mécanique]
Denis Roegel [Assistant Professor, University Nancy 2]

Guest Researchers

Paul Gibson [National University of Ireland at Maynooth, associated researcher (CNRS), until August 31]
Bernhard Schätz [Technical University of Munich, invited professor (INRIA), October 22–November 11]

Post-doctoral Researcher

Arnaud Lanoix [CNRS, joint with DEDALE group, until August 31]

Ph.D. Students

Nazim Benaïssa [UHP Nancy 1]
Loïc Fejoz [UHP Nancy 1, supported by Microsoft ERO grant]
Houda Fekih [joint supervision, University of Tunis and INPL]
Eun Young Kang [UHP Nancy 1]
Olfa Mosbahi [joint supervision, University of Tunis and INPL]
Cyril Proch [UHP Nancy 1, PhD defense on March 22, 2006]
Joris Rehm [UHP Nancy 1, grant of French ministery of Higher Education and Research, since September 2006]
Yann Zimmermann [UHP Nancy 1, PhD defense on November 29, 2006]

Master Students

Clément Hurlin [Master Nancy]
Joris Rehm [Master Nancy]

Student Interns

Aditi Gupta [I.I.T. Kanpur, India, May – July, 2006]
Bo Zhang [TU Munich, Germany, May – July, 2006]

2. Overall Objectives

2.1. Overall Objectives

Proof-oriented system development complements standard methods for the design of computerized systems by formal descriptions and analysis techniques that help to ensure higher levels of reliability and correctness. The MOSEL research team develops such concepts, and applies them, focussing on reactive, real-time, distributed, and mobile systems and that may contain both hardware and software components. Key concepts

in the approach advocated by our group are *refinement* and *(de-)composition* that support the development of complex systems across several layers of abstraction. Our work is structured along the following lines of research:

Foundations and methodology. The theoretical underpinnings of formal methods have been firmly established since several decades, and we refrain from developing completely new approaches. However, novel concepts of system design or novel application domains, including the security of computerized systems, mobile systems or hardware/software codesign require extensions and adaptations of existing formalisms (B and TLA$^+$ are the two main frameworks used by our group). Moreover, formal methods need to be integrated in standard industrial development cycles, requiring serious attention to the methodology of their application. For example, specifications and proofs represent proper artefacts of system design, and we are engaged in work on their representation, management, and reuse, based on composition and genericity.

Notation and tools. We are developing specific notation to aid system engineers represent their concepts and to integrate different methods and tools of system design. Where necessary, we also engage in developing support tools or—preferably—in interfacing existing tools to facilitate their use or support their application in novel contexts.

Applications. Industrial and academic case studies serve to validate our concepts and theories and lay the foundation for their transfer to use by practitioners in industry. They also force us to recognize deficiencies of our concepts, stimulating further theoretical advances and tool development. We are therefore maintaining active cooperations with partners in industry and academia, including neighboring disciplines such as circuit design. We also use our methods in the courses we teach to evaluate their applicability.

The cooperation with research groups within France and elsewhere helps us to clarify and promote our ideas. Within LORIA, we are actively participating in the QSL (*Qualité et Sûreté des Logiciels*) theme of research.

3. Scientific Foundations

3.1. Foundations and Methodology

Keywords: *abstraction*, *composition*, *concurrency*, *distributed systems*, *formal methods*, *reactive systems*, *refinement*.

The MOSEL team investigates methods to develop provably correct computer-based systems. The class of systems we are interested in includes reactive, distributed, embedded, and mobile systems. In contrast to classical sequential algorithms that can be characterized in terms of their input-output relation, the correctness of such systems is described in terms of their executions (traces). The choice of an adequate formal language depends on which properties are of interest for a given system. For example, methods based on pre- and postconditions suffice for expressing and proving safety properties, while temporal logics can also express liveness.

We are particularly interested in processes and methodologies that underly system *development*, as opposed to the verification of an existing system a posteriori. This view is formally reflected by the notion of *refinement*, which ensures that descriptions produced in later stages of system design preserve earlier, more abstract descriptions; in particular, all properties proven earlier remain valid for the refined model. In this way, the effort of verification is spread over the entire development process, and this helps us to achieve a significant degree of automatisation in our verification efforts. Crucially, errors can be detected very early, when they are relatively cheap to correct. The formalisms that we are most familiar with are the B method due to Abrial [46], [45] and the Temporal Logic of Actions and the TLA$^+$ language introduced by Lamport [48]. Members of MOSEL have been invited to write tutorials on these methods [2], [5].

The second cornerstone of system development is composition and decomposition [44]. In effect, monolithic system development methods do not scale to realistic systems. Composition refers to the assembly of complex systems from independently developed, possibly pre-existing components. Dually, an entire system (or its specification) can be decomposed into separate subsystems that are then refined individually. Decomposition is a fundamental structuring principle of the event-based B method.

The contributions of the MOSEL team to the foundations of this area concern extensions of the semantic models for particular types of systems such as real-time, mobile or security-sensitive systems. We also study ways to make developments more easily reusable by focusing on generic theories and proofs that can later be instantiated for reuse.

3.2. Notation and tools

Keywords: *B*, *TLA*, *interface*, *model checking*, *proof obligations*, *theorem proving*, *tool integration*.

The development of provably correct systems relies on languages with a precise, mathematically defined semantics, in which system specifications are written and proof obligations are stated. For all but toy systems, formal development methods generate a huge number of proof obligations, and highly automated tools become essential to successfully apply the methods. Whereas automated deduction has made substantial progress, each tool typically covers a restricted domain, and the combination of different tools is an active area of research.

In this spirit, we introduced the format of *predicate diagrams* [3], [4] to represent Boolean abstractions of reactive systems in the form of finite-state diagrams, with annotations that express fairness and liveness properties. Predicate diagrams form an interface between theorem proving and model checking techniques: the former are useful for showing the correctness of the abstraction, whereas the latter can establish temporal properties from the finite-state diagram representation. The DIXIT tool (described in section 5.2) implements an editor for predicate diagrams, generates proof obligations to show the correctness of abstractions, and invokes external model checkers to verify properties of abstractions expressed in temporal logic. Members of MOSEL have also contributed intensively to the development of the SMT solver haRVey (described in section 5.1) and to its integration with interactive proof assistants (see section 6.7).

We believe that beyond the sheer capacity of provers for carrying out deductions, adequate interfaces are an important element in order to promote their actual use in system development. Dominique Cansell has developed an interface for the interactive prover of Atelier B, the primary support tool for the B method, freely available for academic users within the B4Free tool. Its development is based on a thorough study of interactive provers are used for system verification.

3.3. Applications

Keywords: *access control*, *digital TV*, *embedded systems*, *hardware-software codesign*, *security*, *service interaction*, *services*, *telecommunications*.

A substantial part of our research is driven by work on concrete applications and case studies, as well as by courses that we teach. Several applications currently studied by MOSEL concern hardware-software codesign for embedded systems. Within these industrial projects, we have applied the B method to produce a series of models, starting from standard requirement documents as inputs for the abstract models. As a result of several refinement steps, with accompanying proofs, we were obtained models at a level of granularity that allowed us to mechanically synthesize hardware descriptions.

In the context of two national projects on computer security (see sections 8.5 and 8.6), we apply our methods to the development of services and to access-control applications. In particular, we try to combine logics and formalisms traditionally used for the specification of access control requirements with state-based notations such as B and TLA$^+$.

4. Application Domains

4.1. Application Domains

Keywords: *critical systems*, *embedded systems*, *networks*, *protocols*, *telecommunications*.

Our work mainly targets critical systems whose malfunctioning may endanger the health or life of persons, their privacy and security, or that may lead to serious financial consequences. We enjoy working on concrete examples that are developed in the context of industrial or cooperative projects, including telecommunications, embedded systems, networks and their protocols, problems of information security, and mobile systems.

5. Software

5.1. The haRVey reasoner

Participants: Pascal Fontaine, Stephan Merz.

haRVey is a SMT (Satisfiability Modulo Theories) prover. It is developed in cooperation with Silvio Ranise and Christophe Ringeissen of the CASSIS project team of INRIA Lorraine, and with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brasil. haRVey can handle large quantifier-free formulas containing uninterpreted predicates and functions, and some arithmetics. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about "higher-order" concepts involving sets, relations, etc. The prover can produce an explicit proof trace when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols. This feature has been used to integrate it with the proof assistant Isabelle (see section 6.7).

This year, the tool has been made available at http://harvey.loria.fr. We are currently working on the upcoming version of the tool, which will provide a better integration of the different available features, and a better support for arithmetic on rationals and integers.

Future research and implementation efforts will be directed to furthermore extend the accepted language, increase the efficiency, and provide an optimal interface (including providing explicit proof traces) for the prover to be used within larger verification tools.

We target applications where validation of formulas is crucial, such as validation of TLA$^+$ and B specifications.

5.2. The DIXIT toolkit

Participants: Loïc Fejoz, Dominique Méry, Stephan Merz.

The DIXIT toolkit provides support for the verification of systems using Boolean abstractions in the form of predicate diagrams. It is organized around a visual editor (based on the GEF platform) that allows a user to draw a predicate diagram and enter node and edge annotations. Properties expressed in linear-time temporal logic can be verified from the interface by calling the Spin or LWAASpin model checkers, counter-examples are visualized in the editor, and proof obligations that ensure the correctness of the abstraction can be generated. Finally, the toolkit can verify that a predicate diagram refines a more abstract one, ensuring the preservation of temporal logic properties. Except for occasional bug-fixes, no major new developments have occurred in 2006, but DIXIT is now registered with APP and is freely available for download.

6. New Results

6.1. Incremental development of distributed algorithms

Keywords: *distributed algorithms*, *event B*, *refinement*.

Participants: Dominique Cansell, Jacques Jaray, Dominique Méry, Olfa Mosbahi.

The development of distributed algorithms and, more generally, distributed systems, is a complex, delicate, and challenging process. Refinement techniques of (system) models improve the process by using a proof assistant, and by applying a design methodology aimed at starting from the most abstract model and leading, in an incremental way, to the most concrete model, for producing a distributed solution. Our works help to formalize pre-existing algorithms as well as to develop new algorithms. We have shown, using the distributed reference counting (DRC) problem as our study [14], how models can be produced in an elegant and progressive way, thanks to the refinement and how the final distributed algorithm is built starting from these models. Similarly, Simpson's 4-slot algorithm is a reader/writer algorithm which is lock-free asynchronous: the reader reads the last written value. The reader and the writer work only on four memory cells. Jean-Raymond Abrial and Dominique Cansell have developed this algorithm based on the refinement methodology. The first abstract model designs only the reading and writing trace. The reading trace disapears in the second refined model and the writing trace is substituted by the data structure of Simpson's algorithm. In the last refinement, each event of the writer and of the reader are atomic. This work was presented at a Dagstuhl seminar on Atomicity in March 2006. Finally, Jean-Raymond Abrial, Dominique Cansell, and Dominique Méry have reconsidered the leader election algorithm of the Firewire protocol to obtain a new algorithm; we have proved that acknowlegments and confirmations are not necessary. The protocol works without these redundant messages. This work was presented at a Dagstuhl seminar on Rigorous Methods for Software Construction and Analysis in May 2006.

The doctoral thesis of Olfa Mosbahi is co-supervised by Jacques Jaray and Samir Ben Ahmed from the Institut Supérieur d'Informatique in Tunis. It concerns the joint use of the B and TLA$^+$ methods for modeling and designing systems with temporal requirements. She has proposed to use composition techniques in order to combine models of control systems and is also looking at refinement of real-time systems. Her work has been published at two conferences [28], [29].

6.2. Modelling of electronic systems

Keywords: *B method*, *BHDL*, *hardware description*, *refinement*, *theorem proving*.

Participants: Dominique Cansell, Yann Zimmermann.

As the complexity of electronics systems continues to increase and their reliability requirements become more and more important, the challenge is to master complexity in development while ensuring the correctness of systems. Whereas test-based methods no longer provide sufficient coverage for realistic systems, proof-based methods are not limited by the complexity of systems. In his PhD thesis [9], Yann Zimmermann suggests using the B method and its concept of refinement to simplify the process of modelling and proving. At each refinement step, proof obligations are automatically generated by tools to ensure that the concrete model is correct with respect to the abstract model. This method ensures that the final implementation is correct with respect to the initial abstract specification. The point of departure for this work was a realistic case study chosen by Volvo (SAE J1708). The case study concerns a communication protocol that allows different components of a car to send and receive messages over a shared bus. This case study has led us to define some modelling rules to develop electronic circuits using the B method. We have defined the BHDL language which is a subset of B from which circuits can be synthesized, and we have implemented translators from BHDL towards VHDL and SystemC. The semantics of BHDL has been defined formally, and the correctness of the translation from BDHL to SystemC has been proven. Some work has also been done to translate BHDL to ACL2.

6.3. Formalization of Access Control Specifications

Keywords: *B method*, *access control*, *refinement*.

Participants: Nazim Benaïssa, Dominique Cansell, Dominique Méry, Stephan Merz.

Triggered by interactions in the DESIRS project (see section 8.6), we have worked on augmenting Event-B models by primitives for access control, and in particular permissions, restrictions, and obligations. The idea is to add suitable predicates to the definition of system events, and to verify that the event specification respects these annotations. We have also studied how such annotations can be taken into account by refinements. Restrictions and obligations can be interpreted as safety and liveness properties of traces and are therefore guaranteed to be preserved by the usual refinement notion of trace inclusion. Permissions, however, express the existence of certain branches, and are not in general guaranteed to be preserved. We show that high-level permissions can be preserved by a combination of low-level permissions and obligations, and this observation implies that the correctness can be reduced to standard verification conditions in first-order logic. A paper describing these ideas [31] has appeared at the Intl. Workshop on the Theory of Security.

In [34], we address the proof-based development of (system) models satisfying a security policy. The security policy is expressed as an OrBAC model. This formalism allows a developer to state permissions and prohibitions on actions and activities and belongs to the family of role-based access control formalisms. The main question is to validate the link between the security policy expressed in OrBAC and the resulting system; a first abstract B model is derived from the OrBAC specification of the security policy and then the model is refined to introduce properties that can be expressed in OrBAC. The refinement guarantees that the resulting B (system) model satisfies the security policy. We present a generic development of a system with respect to a security policy; this generic model can then be instantiated for a given security policy.

6.4. Time constraint patterns for event B development

Keywords: *distributed systems*, *event B*, *pattern*, *refinement*.

Participants: Dominique Cansell, Dominique Méry, Joris Rehm.

Real-time constraints are frequent requirements for distributed applications. In order to express such constraints in (mathematical) models, we intend to integrate time constraints in the modelling process based on event B models and refinement. The starting point of our work is the event B development of the IEEE 1394 leader election protocol; from standard documents, we derive temporal requirements to solve the contention problem and we propose a method for introducing time constraints using a pattern. The pattern captures time constraints in a generic event B development and it is applied to the IEEE 1394 case study. In his master's thesis [43], Joris Rehm introduced time constraints (propagation, sleeping time) in the B development of the IEEE 1394 protocol to solve the contention problem. To attend this goal he has defined B pattern to introduce time constraint. A first published paper [32] describing the introduction of time constraint using refinement was published at the AVoCS workshop; an internal report[35] provides new elements on the topic, and a fuller paper has been accepted for publication at B 2007.

6.5. A Proof Language for TLA+

Keywords: *TLA*, *proof language*.

Participant: Stephan Merz.

In the context of a joint project between INRIA and Microsoft Research (see section 8.3) that aims at developing a verification environment for TLA$^+$, we have worked on the definition of a declarative and hierarchical proof language for TLA$^+$. The objective is for proofs to be readable independently of an interactive proof assistant, which is difficult with traditional, tactic-based interaction.

This year, in discussions with Leslie Lamport, Georges Gonthier, and Damien Doligez, we have in particular defined the overall structure of hierarchical proofs, based on a numbering scheme for the high levels of the proof and unnumbered proof sequences at the lower levels. We have identified the basic primitives of the proof languages, as well as some convenient abbreviations that codify standard proof idioms. We have also defined a schema for naming (sub-)expressions and formulas in a proof.

6.6. Modeling and Verification of Domain-Specific Languages

Keywords: *CPL*, *TLA*, *domain-specific language*, *model checking*, *verification*.

Participants: Pascal Fontaine, Dominique Méry, Stephan Merz.

Within the INRIA cooperative research initiative VeriTLA$^+$ (see section 8.4), we have mainly worked with members of the PHOENIX project in Bordeaux on the verification of VisuCom and CPL, two languages used for telephony services. In particular, in work with Julien Mercadal, who visited LORIA in July, we have defined the translation from these languages into TLA$^+$, as well as the expression of (fixed) domain-specific correctness properties. These translations have been implemented as back-ends of the respective compilers. We then use the TLC model checker to verify the correctness of fixed instances of these models, which has enabled us to detect errors that could not be found with static analysis techniques.

It appears that many proof obligations that are generated in this way fall into decidable fragments of first-order logic and could be verified using an SMT solver such as haRVey, and we plan to address this in more detail in future work, as well as extend the approach to full SPL (signaling processing language).

6.7. Cooperation of deductive tools based on proof reconstruction

Keywords: *SMT solvers*, *proof reconstruction*, *proof assistants*, *set theory*, *theorem proving*.

Participants: Pascal Fontaine, Clément Hurlin, Stephan Merz, Leonor Prensa Nieto.

We have continued to work on techniques for combining interactive proof assistants (specifically, Isabelle/HOL) and automated provers (specifically, haRVey) based on the certification within the proof assistant of a proof trace generated by the automatic prover. The work has partly been carried out in a joint project with the Isabelle group at the Technical University of Munich led by Tobias Nipkow and with John Matthews of Galois Solutions, Portland, Oregon (see section 8.7). Our original code for SAT solver integration is now part of the Isabelle standard distribution. For quantifier-free first-order logic with uninterpreted function and predicate symbols, we have continued work on improving the stability and efficiency of proof reconstruction. This work has been published in a paper presented at TACAS 2006 [23].

In his master's thesis, Clément Hurlin continued this approach and extended it to quantified formulae and certain set-theoretic operators. This extension is mainly motivated by the fact that languages such as TLA$^+$ or B rely heavily on set theory. The haRVey prover was already extended to (heuristically) handle quantifiers, and it was shown that these heuristics constitute a decision procedure for a fragment of set theory. In this work, we investigated techniques for reconstructing proofs for this class of formulae. The guiding principle is to perform those steps that do not involve proof search (translation from set theory to quantified formulas over uninterpreted function symbols, Skolemization of quantified formulas) as a pre-processing step within Isabelle. haRVey was extended in order to provide proof traces for quantifier instantiation, and Isabelle reconstructs these proofs, thus certifying the correctness of the cooperation.

Clément Hurlin spent part of his research time with the Isabelle development group in Munich. In particular, he explored an approach to the elimination of set-theoretic operators based on reflection during his stay in Munich, with promising results. A first paper [24] describing this work was published at the workshop AVoCS 2006. Future research will aim at a better integration of this extension within haRVey and within proof reconstruction, and at improving the efficiency of the cooperation with the proof assistant.

6.8. Combining reasoners

Keywords: *SMT solvers*, *combination*, *decision procedures*, *theorem proving*.

Participant: Pascal Fontaine.

This work – in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Brazil – is strongly connected with the present development of the haRVey prover. Traditionally, SMT reasoners are based on the Nelson-Oppen combination scheme, where decision procedures exchange (disjunctions of) equalities. We are currently investigating a much more flexible cooperation framework where reasoners (not just decision procedures) generate generic lemmas and communicate them to the other modules in the cooperation. We believe that such a framework will allow us to handle formulas in a very expressive language, beyond what is possible with the traditional scheme. A preliminary description has been published in [21].

The haRVey tool will be used to validate the idea against proof obligations from actual verification tasks, and to study the overall efficiency. Besides, some theoretical questions need to be addressed. We found that the framework is easily proved to be sound. However, the completeness issues still require an in depth treatment.

6.9. Clock synchronization for the FlexRay protocol

Keywords: *FlexRay*, *clock synchronization*, *theorem proving*, *verification*.
Participants: Leonor Prensa Nieto, Bo Zhang.

Our combinations of interactive and automatic theorem provers are mainly intended for the verification of fault-tolerant and distributed algorithms. As a substantial case study for validating our techniques and tools, we study the verification of parts of the FlexRay protocol. FlexRay is a communication protocol intended for advanced automotive control applications, including "X-by-wire" systems. It is based on a combination of time-triggered and event-triggered communication, in order to provide an optimal compromise between high bandwidth and guaranteed response times. As the notion of time is essential for the media access strategy of the protocol, the correctness of the clock synchronization algorithm is the most basic and most important property of the protocol, and we focus on its verification.

We have formalized in Isabelle/HOL a framework for clock synchronization developed by Schneider in the 1980s, and have mechanically verified the correctness of an abstract model of the Lundelius-Lynch midpoint algorithm, on which FlexRay is based. This work has been accepted for publication in a special issue of the journal *Formal Aspects of Computing* [10]; the underlying theories are available at the Archive of Formal Proofs [39]. We have also worked on an extension of this model for FlexRay, which adds substantial complications. First, time is counted in terms of microticks and macroticks, and the lengths of microticks of different controllers do not have to agree. Second, FlexRay adjusts deviations of local clocks not only by (additive) offset corrections as in the Lundelius-Lynch protocol, but also by (multiplicative) rate corrections. The main difficulty in this work is the lack of documentation and justification of numerous constants that are introduced in the FlexRay standard document.

Another interesting target for verification in FlexRay is the medium access protocol. During his visit at LORIA, Bo Zhang studied and formalized in Isabelle an abstract model of the bus guardian, which protects the communication channel against faulty communication controllers. Several correctness properties that are stated in the FlexRay standard have been verified for this abstract model. A short communication has appeared at AVoCS 2006 [33].

6.10. Type Systems for Security

Keywords: *non-interference*, *security*, *theorem prover*, *type systems*.
Participants: Gilles Barthe, Leonor Prensa Nieto.

Information flow type systems provide an elegant means to enforce confidentiality of programs. In joint work with Gilles Barthe from INRIA-Sophia Antipolis, we have specified, using the proof assistant Isabelle/HOL, an information flow type system for a concurrent language featuring primitives for scheduling. We have shown that well-typed programs are non-interfering for a possibilistic notion of non-interference. The development, which constitutes to our best knowledge the first machine-checked account of non-interference for a concurrent language, takes advantage of the proof assistant's facilities to structure the proofs about different views of the programming language and to identify the relationships among them and the type system. Our language and type system generalize previous work of Boudol and Castellani [47], in particular by including arrays and lifting several convenient but unnecessary conditions in the syntax and type system.

This work has led to a publication in the Journal of Computing Security [11].

6.11. Modeling and Verification of an E-voting Interface

Participants: Dominique Cansell, Paul Gibson, Aditi Gupta, Dominique Méry.

Electronic voting systems have been subject to numerous severe flaws, and we believe that they constitute a prime example for the application of formal methods. In particular, the requirements of these systems and their interfaces need to be rigorously defined. As a first step in this direction, we have developed formal requirement models for e-voting systems using the B method. A paper has been presented at the First International Workshop on Formal Methods for Interactive Systems (FMIS 2006) [19].

In related work, Margaret McGaley visited LORIA for two weeks in March, and Paul Gibson performed a critical analysis of the recommendations on e-voting by the European Council, which was published at the Usenix Electronic Voting Technology Workshop [27].

During her work on this project, Ms Gupta evaluated a number of advanced software engineering techniques for rigorous development associated with critical systems – automated testing, reverse engineering (from Java code to UML), design by contract (with Java Modelling Language JML). Central to this evaluation was a non-trivial implementation of an e-voting system prototype which evolved as requirements changed.

This prototype will be used in collaboration with a PhD student in Ireland and will result in a technical report and possible conference or workshop publication in 2007.

6.12. Refinement and verification of component-based systems

Keywords: *LTL properties*, *component-based systems*, *refinement*, *verification*.

Participants: Houda Fekih, Arnaud Lanoix, Stephan Merz.

In order to address the verification of large systems, compositional approaches postpone in part the problem of combinatorial explosion during model exploration. We propose to establish a compositional framework in which the verification may proceed through a refinement-based specification and a component-based verification approaches: first, we introduce a constraint synchronised product operator on which to base the automated compositional verification of a component-based system refinement relation. Secondly, safety LTL properties of the whole system are checked from local safety LTL properties of its components. The main advantage of our specification and verification approaches is that LTL properties are preserved through composition and refinement. This joint work with Olga Kouchnarenko of the LIFC laboratory, Besançon has resulted to a paper which has been accepted for publication at the Sixth International conference of Perspectives of System Informatics (PSI'06) in Novosibirsk, Russia [25] (also available as INRIA research report [36]).

In joint work with the DEDALE team of LORIA (particulary Dieudonné Okalas Ossami and Jeanine Souquières), we are studying an operator-based approach for the incremental development of UML diagrams, in particular, the UML 2.0 Protocol State Machines (PSM). We propose some development patterns that preserve notions of conformance inspired from refinement and specification matchings. Each pattern makes evolve the PSM to another one with respect to conformance between the development steps. This work was presented at the Workshop "Trustworthy Software" in Saarbrücken, Germany [26]. A previous research report about building conformant Protocol State Machine is also available [37].

In collaboration with Inès Mouakher and Jeanine Souquières of the DEDALE team of LORIA, we have worked about the adaptation of component interfaces. In component-based software development approaches, components are considered as black boxes. They are only described by their interfaces expressing their visible behaviors. They must be connected in an appropriate way, through their required and provided interfaces. In the best cases, the provided interfaces are checked compatible with the corresponding required interfaces, but in general cases, adapters have to be introduced to connect them. They connect the required operations and attributes to the required ones.interoperability. Compatibility concerns the interface signatures, behavioral aspects and protocol level. We propose to specify component interfaces in B in order to verify

the interoperability. The use of B assembling and refinement mechanisms allows us to express the adapter and to verify the interoperability. This work was presented at the 11th International Workshop on Component Oriented Programming (WCOP'06), during the ECOOP 06 conference, in Nantes, France [30]. A research report on this subject is also available [38].

In a related line of research, Houda Fekih has investigated techniques for the representation of B models by UML diagrams, in order to better understand the structure of the model and to communicate with application developers who are not in general experts in formal modeling. In 2006, she has focused on the derivation of state machines from the operations of a B model. She has also considered the representation of refined models in UML. This work has been published at the ACM Symposium for Applied Computing [22].

6.13. Provably correct lock-free data structure

Keywords: *data structure*, *lock-free*, *proofs*.

Participants: Loïc Fejoz, Stephan Merz.

The development of multi-threaded programs is no longer restricted to operating system experts and specialized application areas, but is entering mainstream programming. A central problem is to ensure that different threads can access shared data structures without interference. The traditional solutions are based on locks, which can be either coarse-grained (applying to the entire data structure) or fine-grained (applying to individual elements of the data structure). Both cases have severe drawbacks: coarse-grained locks limit the possible degree of parallelism, which fine-grained locks are hard to control and prone to deadlocks. Several mechanisms of concurrent programming without locks have been proposed in the literature. In this project, funded by a Microsoft ERO PhD grant and carried out in cooperation with Tim Harris from Microsoft Research Cambridge, we investigate verification techniques for lock-free data structures. We have compared the formalization of such algorithms in the B and TLA$^+$ methods and the application of assumption-guarantee style reasoning, identifying the strong and weak points of both approaches. We intend to develop dedicated program transformation techniques that are tailored to this class of algorithms and that would allow a developer to verify them with a high degree of automation.

6.14. Abstraction-based verification for real-time systems

Keywords: *abstraction*, *real-time systems*, *refinement*, *timed automata*.

Participants: Eun-Young Kang, Stephan Merz.

We study the use of abstraction techniques for the verification of real-time systems, combining predicate abstraction, theorem proving, and model checking, and their application in a methodology called IRA (Iterative Abstract-Refinement Algorithm). Technically, we are using a variant of predicate diagrams adapted to timed systems, that distinguish between discrete and time-passing transitions. IRA helps us identify appropriate predicates on which the abstraction is based. We have demonstrated that automatic tools such as SMT solvers are useful in computing an abstract model of a given system specification, which helps to improve the degree of automation.

This work has been accepted for publication in a special issue of the journal *Formal Aspects of Computing* [15].

6.15. Miscellaneous

Keywords: *Babbage*, *Metapost*, *difference engine*, *graphics*, *history of computing*, *layout*.

Participant: Denis Roegel.

Denis Roegel has continued to develop extensions to METAPOST addressing various abstract representations of objects. He has in particular been studying the correct representation of spheres, with their great circles and parallels, after having observed that almost all such depictions in the literature were contradictory. An article describing this work has been submitted.

He is also completing two chapters of the 2nd edition of the book *LaTeX Graphics Companion*, which is due in the Spring of 2007.

During a visit to Oxford, Denis Roegel was able to analyze one of the few remaining fragments of Charles Babbage's first difference engine. By comparison with other such fragments (Cambridge, London, Sydney, Harvard), it appears that the Oxford fragment contains parts which belong to a different machine, perhaps the prototype built by Babbage around 1821. Our findings have been submitted to the IEEE Annals of the history of computing.

7. Contracts and Grants with Industry

7.1. Microsoft ERO PhD Grant

Participants: Loïc Fejoz, Stephan Merz.

The PhD thesis of Loïc Fejoz (see section 6.13) is supported by a Microsoft ERO PhD grant from January 2006 to December 2008. Our main contact at Microsoft Research is Tim Harris who develops algorithms for lock-free data structures that we intend to verify.

8. Other Grants and Activities

8.1. QSL Operation Think'n'Play

Keywords: *Event B method*, *Proof*, *Proof-based Pattern*, *Refinement*.

Participants: Nazim Benaïssa, Dominique Cansell, Dominique Méry, Joris Rehm.

As part of our ongoing involvement in the QSL theme (Qualité et Sûreté des Logiciels) at LORIA, the MOSEL team has again proposed specific research actions. The operation Think'n'Play was accepted by the QSL council in March. Jean-Raymond Abrial's team at ETH Zürich is our partner for this work. Its main objectives are:

- Define and develop some B design patterns. For us, a B design pattern is a B development which can be reused to produce mechanically some refinement in another B development. Using patterns the proof process can be easier.
- Use our B patterns on specific examples.
- Develop a tool to manipulate B patterns (QSL platform and RODIN platform).

This operation supports the research described in sections 6.1 and 6.4. In April an half day on pattern was organized at Nancy.

8.2. QSL Operation InSpain

Keywords: *FlexRay clock synchronization protocol*, *combination of proof tools*.

Participants: Leonor Prensa Nieto, Pascal Fontaine, Stephan Merz.

The operation InSpain was accepted in June by the council of the QSL (Qualité et Sûreté des Logiciels) theme at LORIA. Its main objectives are:

- the design and implementation of cooperative reasoning between the interactive theorem prover Isabelle and automatic provers, in particular SMT solvers such as haRVey (see sections 6.8 and 6.7), and
- the verification of fragments of the FlexRay protocol with the help of the combination of interactive and automatic proof tools mentioned above. More precisely, we aim at verifying the correction of the clock synchronization algorithm implemented in FlexRay (see also section 6.9).

8.3. Tools and Methodologies for Formal Specifications and for Proofs

Participant: Stephan Merz.

As part of the Joint Laboratory between INRIA and Microsoft Research, Stephan Merz participates in the project on Tools and Methodologies for Formal Specifications and for Proofs that has started in January, 2006. The objective of the project is to develop an environment for modeling and verifying distributed algorithms in TLA$^+$ (see also 6.5).

8.4. CRI VeriTLA+

Participants: Pascal Fontaine, Dominique Méry, Stephan Merz.

Our team cooperates with the INRIA projects GALLIUM (Damien Doligez), OBASCO (Gilles Muller) at EMN Nantes, and PHOENIX (Charles Consel) as well as with Julia Lawall of DIKU Copenhague and Leslie Lamport of Microsoft Research in the Cooperative Research Initiative VeriTLA$^+$ (2006/07). The objective of this project is to contribute to the development of a TLA$^+$ verification environment and, more specifically, to study the verification of domain-specific languages, in particular for operating system schedulers and for multimedia and telephony services.

8.5. Project CORSS

Keywords: *composition*, *interaction of services*, *refinement*, *resource allocation*, *system kernels*, *system services*, *verification*.

Participants: Dominique Cansell, Dominique Méry, Stephan Merz, Leonor Prensa Nieto.

The CORSS (Composition et raffinement de systèmes sûrs) project was accepted in 2003 within the ACI Sécurité Informatique, a French program for information security. It includes participants with a background in system development and others with a focus on formal methods. Its objective is to study and apply methods and techniques for the development of provably correct systems (or system services) whose specification includes security properties. Our partners for this project are the OBASCO project at EMN Nantes (Gilles Muller), the IRIT Toulouse (Jean-Paul Bodeveix and Mamoun Filali), the PHOENIX project of INRIA at Bordeaux (Charles Consel) and the ARLES project at INRIA Rocquencourt (Valérie Issarny). We have worked on case studies provided by ARLES, OBASCO, and PHOENIX concerning group establishment protocols for mobile ad-hoc networks and domain-specific languages for operating system kernels and telephony services.

8.6. Project DESIRS

Keywords: *B method*, *deontic logic*, *refinement*, *security*.

Participants: Nazim Benaïssa, Dominique Cansell, Dominique Méry, Stephan Merz.

The DESIRS project (Développement de systèmes informatiques par raffinement des contraintes sécuritaires) was accepted in 2003 within the ACI Sécurité Informatique, a French program for information security. Its topic are logics and mechanisms that permit the description and development of systems that must conform to security policies, standards or regulations. Our partners in this project are the LISI/ENSMA of the University of Poitiers (Yamine Aït-Ameur), the ENST Bretagne (Frédéric Cuppens), the IRIT Toulouse (Philippe Balbiani) and the CRIL Lens (Salem Benferhat). For a description of the results of this project in 2006, see section 6.3.

8.7. Cooperation with the Isabelle group in Munich

Participants: Pascal Fontaine, Stephan Merz, Leonor Prensa Nieto.

Funded by the Centre de Coopération Universitaire Franco-Bavarois, we have worked with the Isabelle group at the Technical University in Munich led by Tobias Nipkow and with John Matthews of Galois Solutions in Portland, Oregon, on the combination of Isabelle/HOL with automatic provers, and in particular haRVey (see section 6.7).

8.8. Exchanges with Tunisia

Participants: Houda Fekih, Jacques Jaray, Dominique Méry, Stephan Merz, Olfa Mosbahi.

We have had a very fruitful cooperation with the team led by Professor Samir Ben Ahmed at the Institut Supérieur d'Informatique (ISI) in Tunis for several years. Jacques Jaray was invited several times to teach courses at the Master's level. The cooperation has been reinforced by support from CMCU (*Comité Mixte de Coopération Universitaire*). Currently, Houda Fekih and Olfa Mosbahi are enrolled in a joint Ph.D. program between the University of Tunis and the INPL at Nancy. Olfa Mosbahi has obtained a part-time teaching position at INPL Nancy, with a lighter teaching load than in Tunis.

9. Dissemination

9.1. Program committees and conference organisation

- Arnaud Lanoix was a member of the review committee for the national Conference MAJECSTIC'06 (MAnifestation des JEunes Chercheurs STIC) 2006.
- Stephan Merz served on the program committees of the Software Verification track of the Second Intl. Conf. on Intelligent Computer Communication and Processing (ICCP'06), of the workshop Verify'06 at FLoC 2006, and of the workshop on Trustworthy Software in May in Saarbrücken. Together with Tobias Nipkow, he was program chair of AVoCS 2006, which took place at LORIA on September 18 and 19, 2006 [7].
- Dominique Cansell was a member of the program committees of AFADL 2006 and B2007.
- Dominique Méry was a member of the program committees of FM 2006 and PSI 2006.

9.2. Tutorials, invited talks, panels

- Pascal Fontaine gave a tutorial on the combination of reasoners within the haRVey tool at the conference ICTAC 2006.
- Paul Gibson gave invited talks on requirements and verification of e-voting systems at Clemson University, South Carolina, and at the Workshop on Trustworthy Software organized at Saarbrücken.
- Stephan Merz gave an invited talk at the 40th anniversary meeting of IFIP WG 2.2 in Udine, Italy, in September.
- Denis Roegel contributed a review of the book *Mathematical Illustrations* by Bill Casselman, for the Notices of the AMS [18]. The review is slated to appear in January 2007.
- Dominique Méry gave a tutorial on the event B method at FORTE 2006 organized in Paris in September 2006.

9.3. Theses, habilitations, academic duties

- Dominique Cansell is coordinator of the French B working group of the new research group within CNRS named GPL (leader Yves Ledru).
- Jacques Jaray was dean of studies, as well as acting director of the Ecole des Mines de Nancy, until the beginning of 2006.
- Jacques Jaray has been charged, by the President of INPL, of the design of the INPL graduate school catalogue in order to attract excellent foreign students.
- Dominique Méry is a member of the IFIP Working Group 1.3 on *Foundations of System Specification*.

- Dominique Méry is the Head of the master programme in computer science for the three universities of Nancy.
- Dominique Méry is a member of the scientific council of the University Henri Poincaré Nancy 1.
- Dominique Méry is a member of the scientific council of the LORIA laboratory.
- Dominique Méry is an expert for the French Ministery of Education (DS9).
- Stephan Merz wrote a report on the PhD thesis of Heikki Tauriainen at the Helsinki University of Technology.
- Stephan Merz is the deputy director of the QSL research theme at LORIA.
- Stephan Merz is the delegate for international relations at LORIA and INRIA Lorraine and a member of the managing board of LORIA.
- Stephan Merz is a member of the sub-group on initiative actions of the Conseil d'orientation scientifique et technologique (COST) of INRIA.
- Stephan Merz is an elected member of the evaluation council of INRIA.
- Stephan Merz is a member of the IFIP Working Group 2.2 *Formal Description of Programming Concepts*.
- Stephan Merz is an advisor for the European project Protocure.

9.4. Teaching

The majority of the members of the MOSEL team are employed on university positions and have significant teaching obligations. We only indicate the graduate courses they have been teaching in 2006.

- Dominique Cansell taught a course on the B prover at the summer school for young researchers in programming (EJCP 2006)
- Dominique Cansell and Dominique Méry gave a course in the Master's program at Nancy on the specification and modelling of computer-based systems.
- Jacques Jaray has created a new course in data modelling: advanced data bases and semi-structured data, XML and Java interfaces as well as a course on JavaSpaces.
- Dominique Méry gives courses on the specification of computer-based systems at the Ecole Supérieure d'Electricité of Metz. He is director of international relations at ESIAL Nancy.
- Stephan Merz gave courses on algorithmic verification (together with Olivier Bournez) and on the semantics of parallel and distributed systems at the Master's program in Nancy.
- Leonor Prensa Nieto was invited to present courses on "Theorem Proving with Isabelle/HOL" at the Universitat Politècnica de Catalunya, Barcelona, in February and at the Universidad del País Vasco, San Sebastián, in July.
- Denis Roegel taught a graduate course on scientific typesetting for prospective PhD students of Nancy 1 University. He is also involved in teaching a graduate course on operational research at the CNAM Open University.

10. Bibliography

Major publications by the team in recent years

[1] J.-R. ABRIAL, D. CANSELL, D. MÉRY. *A Mechanically Proved and Incremental Development of IEEE 1394 Tree Identify Protocol*, in "Formal Aspects of Computing", vol. 14, n^{o} 3, 2003, p. 215-227.

[2] D. CANSELL, D. MÉRY. *Foundations of the B method*, in "Computers and Informatics", vol. 22, n^{o} 3–4, 2003, p. 221–256.

[3] D. CANSELL, D. MÉRY, S. MERZ. *Predicate Diagrams for the Verification of Reactive Systems*, in "2nd Intl. Conf. on Integrated Formal Methods (IFM 2000), Dagstuhl, Germany", Lecture Notes in Computer Science, vol. 1945, Springer-Verlag, November 2000, p. 380–397.

[4] D. CANSELL, D. MÉRY, S. MERZ. *Diagram Refinements for the Design of Reactive Systems*, in "Journal of Universal Computer Science", vol. 7, n^{o} 2, 2001, p. 159–174.

[5] S. MERZ. *On the Logic of TLA+*, in "Computers and Informatics", vol. 22, n^{o} 3–4, 2003, p. 351–379.

[6] S. MERZ. *A More Complete TLA*, in "FM'99: World Congress on Formal Methods, Toulouse, France", J. WING, J. WOODCOCK, J. DAVIES (editors). , Lecture Notes in Computer Science, vol. 1709, Springer-Verlag, 1999, p. 1226–1244.

Year Publications

Books and Monographs

[7] S. MERZ, T. NIPKOW (editors). *Automated Verification of Critical Systems (AVoCS'06)*, INRIA Lorraine & LORIA, Nancy, France, 2006.

Doctoral dissertations and Habilitation theses

[8] C. PROCH. *Assistance au développement incrémental et prouvé de systèmes enfouis*, Ph. D. Thesis, Université Henri Poincaré Nancy 1, March 2006.

[9] Y. ZIMMERMANN. *Modélisation et développement formel de systèmes électroniques*, Ph. D. Thesis, Université Henri Poincaré (Nancy 1), Nov 2006.

Articles in refereed journals and book chapters

[10] D. BARSOTTI, L. PRENSA NIETO, A. TIU. *Verification of Clock Synchronization Algorithms: Experiments on a combination of deductive tools*, in "Formal Aspects of Computing", to appear, 2006, http://hal.inria.fr/inria-00097383/en/.

[11] G. BARTHE, L. PRENSA NIETO. *Secure Information Flow for a Concurrent Language with Scheduling*, in "Journal of Computer Security", to appear, 2006, http://hal.inria.fr/inria-00097395/en/.

[12] D. CANSELL. *B Method*, in "The Seventeen Provers of the World", F. WIEDIJK (editor). , Lecture Notes in Computer Science / LNAI, vol. 3600, Springer, 2006, p. 142-150, http://hal.inria.fr/inria-00096700/en/.

[13] D. CANSELL, D. MÉRY. *Event B*, in "Software Specification Methods", H. HABRIAS, M. FRAPPIER (editors). , Hermes-Science Lavoisier, 2006, http://hal.inria.fr/inria-00096696/en/.

[14] D. CANSELL, D. MÉRY. *Formal and Incremental Construction of Distributed Algorithms: On the Distributed Reference Counting Algorithm*, in "Theoretical Computer Science", vol. 364, n^{o} 3, 2006, p. 318-337, http://hal.inria.fr/inria-00093164/en/.

[15] E. KANG, S. MERZ. *Predicate Diagrams for the Verification of Real-Time Systems*, in "Formal Aspects of Computing", to appear, 2006, https://hal.inria.fr/inria-00112065/en/.

[16] A. KNAPP, S. MERZ, M. WIRSING, J. ZAPPE. *Specification and Refinement of Mobile Systems in MTLA and Mobile UML*, in "Theoretical Computer Science", vol. 351, n° 2, 2006, p. 184-202, http://hal.inria.fr/inria-00000754/en/.

[17] S. MERZ. *Model checking : éléments de base*, in "Systèmes Temps Réel - techniques de description et de vérification", N. NAVET (editor). , Hermes-Science Lavoisier, 2006, p. 89-120, http://hal.inria.fr/inria-00081343/en/.

[18] D. ROEGEL. *Review of "Mathematical Illustrations: A Manual of Geometry and PostScript"*, in "Notices of the American Mathematical Society", vol. 54, n° 1, 2006, http://hal.inria.fr/inria-00111243/en/.

Publications in Conferences and Workshops

[19] D. CANSELL, J. P. GIBSON, D. MÉRY. *Refinement: a constructive approach to formal software design for a secure e-voting interface*, in "Formal Methods for Interactive Systems (FMIS 2006), Macao", 2006.

[20] D. CANSELL, D. MÉRY. *Incremental Parametric Development of Greedy Algorithms*, in "Automatic Verification of Critical Systems - AVoCS 2006, Nancy, France", S. MERZ, T. NIPKOW (editors). , 2006, p. 48-62, http://hal.inria.fr/inria-00089497/en/.

[21] D. DÉHARBE, P. FONTAINE. *haRVey: combining reasoners*, in "Automatic Verification of Critical Systems - AVoCS 2006, Nancy, France", S. MERZ, T. NIPKOW (editors). , 2006, p. 152-156, http://hal.inria.fr/inria-00091662/en/.

[22] H. FEKIH, L. JEMNI BEN AYED, S. MERZ. *Transformation of B Specifications into UML Class Diagrams and State Machines*, in "21st Annual ACM Symposium on Applied Computing - SAC 2006, Dijon, France", vol. 2, ACM, 2006, p. 1840-1844, http://hal.inria.fr/inria-00001269/en/.

[23] P. FONTAINE, J.-Y. MARION, S. MERZ, L. PRENSA NIETO, A. TIU. *Expressiveness + Automation + Soundness: Towards Combining SMT Solvers and Interactive Proof Assistants*, in "12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems - TACAS'06, Vienna, Austria", H. HERMANNS, J. PALSBERG (editors). , Lecture Notes in Computer Science, vol. 3920, Springer, 2006, p. 167-181, http://hal.inria.fr/inria-00001088/en/.

[24] C. HURLIN. *Proof reconstruction for first-order logic and set-theoretical constructions*, in "Automatic Verification of Critical Systems - AVoCS 2006, Nancy, France", S. MERZ, T. NIPKOW (editors). , 2006, p. 157-162, http://hal.inria.fr/inria-00091811/en/.

[25] O. KOUCHNARENKO, A. LANOIX. *How to Verify and Exploit a Refinement of Component-based Systems*, in "Perspectives Of System Informatics (Sixth A. Ershov Memorial Conf., PSI'06), Novosibirsk, Russia", I. VIRBITSKAITE, A. VORONKOV (editors). , Lecture Notes in Computer Science, to appear, vol. 4378, Springer, 2006, p. 457–469, http://hal.inria.fr/inria-00110527.

[26] A. LANOIX, D.-D. OKALAS OSSAMI, J. SOUQUIÈRES. *An Operator-based Approach to Incremental Development of Conform Protocol State Machines*, in "Trustworthy Software, Dagstuhl, Germany", Inter-

nationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Saarland University, April 2006, http://drops.dagstuhl.de/opus/volltexte/2006/695/.

[27] M. MCGALEY, J. P. GIBSON. *"A Critical Analysis of the Council of Europe Recommendations on e-voting"*, in "USENIX/ACCURATE Electronic Voting Technology Workshop", 2006.

[28] O. MOSBAHI, J. JARAY, L. JEMNI BEN AYED. *A Formal Development Method of Control Systems using Event B Approach*, in "4th ACS/IEEE International Conference on Computer Systems and Applications, Dubai", 2006, http://hal.inria.fr/inria-00102208/en/.

[29] O. MOSBAHI, L. JEMNI BEN AYED, J. JARAY. *Formal Development Method of Automated Systems using the Temporal Logic of Actions TLA*, in "6ième Conférence Francophone de Modélisation et Simulation des Systèmes - MOSIM'06, Rabat", 2006, http://hal.inria.fr/inria-00102229/en/.

[30] I. MOUAKHER, A. LANOIX, J. SOUQUIÈRES. *Component Adaptation: Specification and Verification*, in "11th Intl. Workshop on Component Oriented Programming (WCOP 2006), Nantes", Ecole des Mines, July 2006, p. 23-30, http://hal.inria.fr/inria-00074477.

[31] D. MÉRY, S. MERZ. *Event Systems and Access Control*, in "Sixth International IFIP WG 1.7 Workshop on Issues in the Theory of Security, Vienna, Austria", D. GOLLMANN, J. JÜRJENS (editors). , Vienna University of Technology, 2006, p. 40-54, http://hal.inria.fr/inria-00001262/en/.

[32] J. REHM. *A method to refine time constraints in event B framework*, in "Automatic Verification of Critical Systems - AVoCS 2006, Nancy, France", S. MERZ, T. NIPKOW (editors). , September 2006, p. 173-177, http://hal.inria.fr/inria-00091665/en/.

[33] B. ZHANG. *On the Formal Verification of the FlexRay Communication Protocol*, in "Automatic Verification of Critical Systems - AVoCS 2006, Nancy, France", S. MERZ, T. NIPKOW (editors). , 2006, p. 184-189, http://hal.inria.fr/inria-00091662/en/.

Internal Reports

[34] N. BENAÏSSA, D. CANSELL, D. MÉRY. *Integration of security policy into system modeling*, Technical report, LORIA, October 2006.

[35] D. CANSELL, D. MÉRY, J. REHM. *Time constraint patterns for event B development*, Technical report, LORIA, October 2006.

[36] O. KOUCHNARENKO, A. LANOIX. *How to Verify and Exploit a Refinement of Component-based Systems*, Rapport de recherche, n^{o} RR-5898, INRIA, 2006, http://hal.inria.fr/inria-00071369/en/.

[37] A. LANOIX, J. SOUQUIÈRES. *A Step-by-step Process to Build Conform UML Protocol State Machines*, Research Report, LORIA, February 2006, http://hal.archives-ouvertes.fr/hal-00019314.

[38] A. LANOIX, J. SOUQUIÈRES. *Component-based Development using the B method*, Research Report, LORIA, July 2006, http://hal.archives-ouvertes.fr/hal-00105041.

Miscellaneous

[39] D. BARSOTTI. *Instances of Schneider's Generalized Protocol of Clock Synchronization*, 2006, http://afp.sourceforge.net/entries/ClockSynchInst.shtml, Archive of Formal Proofs.

[40] L. FEJOZ. *DIXIT*, APP IDDN.FR.001.150022.000.R.P.2006.000.10600, 2006, http://www.loria.fr/equipes/mosel/research/dixit/.

[41] P. FONTAINE. *Harvey-Sat*, APP IDDN.FR.001.220025.000.S.P.2006.000.10000, 2006, http://harvey.loria.fr.

[42] C. HURLIN. *Reconstruction de preuves pour les formules quantifiées et ensemblistes*, Technical report, Université Henri Poincaré Nancy 1, Nancy, 2006.

[43] J. REHM. *Contraintes temporelles dans les modèles événementiels*, Technical report, Université Nancy 2, Nancy, 2006.

References in notes

[44] W.-P. DE ROEVER, H. LANGMAACK, A. PNUELI (editors). *Compositionality: The Significant Difference*, Lecture Notes in Computer Science, vol. 1536, Springer-Verlag, 1998.

[45] J.-R. ABRIAL. *Extending B without changing it (for developing distributed systems)*, in "1st Conference on the B method", H. HABRIAS (editor). , IRIN Institut de recherche en informatique de Nantes, 1996, p. 169–190.

[46] J.-R. ABRIAL. *The B-Book: Assigning Programs to Meanings*, Cambridge University Press, 1996.

[47] G. BOUDOL, I. CASTELLANI. *Noninterference for Concurrent Programs and Thread Systems*, in "Theoretical Computer Science", vol. 281 (Merci, Maurice. A mosaic in honour of Maurice Nivat), n^{o} 1, 2002, p. 109–130.

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