INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Mosel

Proof-oriented development of computer-based systems

Lorraine

THEME SYM

Activity Report

2005

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 belongs to the faculty of the University of Metz.

Team Leader

Dominique Méry [Professor, University Henri Poincaré Nancy 1]

Administrative Assistant

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

INRIA Staff

Stephan Merz [Research Director, INRIA]

University Staff

Dominique Cansell [Assistant Professor, University of Metz]
Jacques Jaray [Professor, Institut National Polytechnique de Lorraine and École des Mines de Nancy]
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), since September 1]
Bernhard Schätz [Technical University of Munich, invited professor (INRIA), September 24–October 22]

Post-doctoral Researcher

Pascal Fontaine [INRIA, until September 30; ATER Nancy 2, since October 1]
Arnaud Lanoix [CNRS, joint with DEDALE group, since October 1]
Alwen Tiu [QSL group, until October 31]

Junior technical staff

Loïc Fejoz [Ministry of Education and Research, ACI Sécurité Informatique]

Ph.D. Students

Houda Fekih [joint supervision, University of Tunis, until 2007]
Eun Young Kang [UHP Nancy 1, until 2007]
Olfa Mosbahi [joint supervision, University of Tunis, until 2007]
Cyril Proch [UHP Nancy 1, until February 2006]
Yann Zimmermann [UHP Nancy 1, until 2006]

Master Students

Nazim Benaïssa [DEA Informatique]
Loïc Fejoz [DEA Informatique]

Student Interns

Niket Agarwal [I.I.T. Kharagpur, India, May 1–July 31]
Damian Barsotti [FAMAF, Univ. Nacional de Córdoba, Argentina, April 1–June 30]
Kamal Kant Gupta [I.I.T. Guwahati, India, May 1–July 31]
Mauro Jaskelioff [FCEIA, Univ. de Rosario, Argentina, April 1–June 30]

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, often allowing for a higher degree of automatisation. 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 [32], [31] and the Temporal Logic of Actions and the TLA$^+$ language introduced by Lamport [36].

The second cornerstone of system development is composition and decomposition [30]. 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 (see sections 6.14 and 8.2). We also study ways to make developments more easily reusable by focusing on generic theories and proofs that can later be instantiated for reuse. Work on the RNRT EQUAST project (see section 7.1) opened a new, interesting avenue: the application of formal models to a technical standard document allowed us to identify the underlying hierarchy of parameters. This now only helped explain designs that have traditionally relied on the intuition of experienced engineers, but has resulted in the design of a system on chip that is correct by construction (see also section 6.4).

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.

Our team 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 (see the DIXIT tool in section 5.2). As part of our implication in the QSL project, we have also worked on combinations of interactive proof assistants and SAT and SMT solvers (see 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 (see section 5.1), now freely available for academic users within the B4Free tool. Its development is based on a thorough study of how humans use an interactive prover.

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 (see section 7.1). 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 section 8.2), 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, descriptions of secure systems, and mobile systems.

5. Software

5.1. Click'n'Prove

Participant: Dominique Cansell.

In cooperation with Jean-Raymond Abrial, and as a result of the QSL operation AdHoc (2001–2003), the interface Click'n'Prove (or "Balbulette" in French) was developed for the B prover, the principal interactive proof support for the B method. This interface, based on (X)Emacs, has been designed to be ergonomic and to guide the user in carrying out a verification task. It has been available as part of the B4free toolset since February, 2004, and has been downloaded by more than 300 subscribers worldwide. An improved version 2.0 has been released in 2005 and was presented at the tool session of the conference ZB 2005. In particular, the new version supports mathematical symbols based on the XSymbols mode.

5.2. DIXIT

Participants: Niket Agarwal, Nazim Benaïssa, Loïc Féjoz, 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 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 (see also section 6.15). 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.

DIXIT has been extended by new functionality, such as the generation of proof obligations in TLA$^+$, B, and MathML formats, and several bugs have been fixed. It is available for download from our Web site and has been registered with the APP. A paper describing the tool has been published at AVIS 2005 [18].

During his internship at LORIA, Niket Agarwal developed experience for adding new functions to DIXIT and introduced techniques for computing predicate diagrams from B models. He wrote functions for translating B models into TLA$^+$ modules.

5.3. haRVey

Participant: Pascal Fontaine.

haRVey is a SMT (Satisfiability Modulo Theories) prover. It is developed in cooperation with the CASSIS project team of INRIA Lorraine. 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 about 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 only. This feature has been used to integrate it with the proof assistant Isabelle (see section 6.7).

Current implementation efforts in haRVey are targeted to 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 extending the accepted language, increasing its efficiency, and providing an optimal interface (including providing explicit proof traces) for the prover to be used within larger verification tools.

5.4. SO6 Plugin for Netbeans

Participant: Loïc Fejoz.

Several software development projects in our team make use of the Java IDE Netbeans. In the future we would like to use Libresource and the synchronizer So6 as a repository for our projects. So6 is a safe and generic synchronizer developed at LORIA by the ECOO team, intended as a replacement for CVS.

We have produced a Netbeans plugin that enables us to directly use So6 commands from inside the editor. The So6 Plugin for Netbeans 3.6 is currently hosted on a Libresource server. It will be freely available after an impending APP deposit.

6. New Results

6.1. Refinement and Reachability in Event-B

Keywords: *B method*, *reachability*, *refinement*, *stuttering*.

Participants: Dominique Cansell, Dominique Méry.

Since the early 90s (following seminal work by R. Back), the refinement of stuttering steps is performed by means of new actions (called here events) refining skip. In joint work with J.-R. Abrial, we show that such an approach to refinement is not always possible in the development of large systems. We shall instead use events refining some kind of non-deterministic actions maintaining the invariant (sometimes called keep). We show that such new refinements are completely safe. In a second part, we explain how such a mechanism can be used to express some reachability conditions that were otherwise expressed using some special temporal logic statements à la TLA. We illustrate our proposal with several examples. This work was presented at ZB 2005 [13] where it received the best paper award.

6.2. Incremental development of parallel and distributed algorithms

Keywords: *B method*, *distributed systems*, *parallel algorithms*, *refinement*.

Participants: Dominique Cansell, Dominique Méry.

In joint work with J.-R. Abrial, we investigate the applicability of the Event-B method for parallel and distributed systems. In particular, the design of non-blocking concurrent algorithms has attracted much attention recently, because it promises better scalability and performance than conventional algorithms based on centralized control.

We present a formal (and mechanically proved) development of a non-blocking and concurrent algorithm by M.M. Michael and M.L. Scott. This algorithm is executed in a highly concurrent fashion by an unknown number of independent processes. These processes access a shared queue by dequeueing the first value and/or enqueueing a new value in the queue. Every process progresses on atomic actions (some of them use a Compare and Swap instruction) so it is extremely difficult to assure the correctness of this algorithm. Using refinement we discover some critical issues and solve them (including the so-called “ABA” problem). We have also found some errors in one guard of the algorithm and corrected them.

This work was presented in a QSL day on “Proving Pointer Programs” and in a Dagstuhl Seminar on atomicity [7]. We plan to extend this work and develop a methodology for the development of provably correct, non-blocking algorithms.

In another case study, we give a proof-based and formal development of a distributed reference counting algorithm. This algorithm is used to propagate resources in a distributed system and also in distributed garbage collection. A resource is created by a site (the owner) and can be used by other sites. The owner of a resource can remove it only when it is sure that this resource is not used by another site. L. Moreau has developed such an algorithm and has proved it with J. Duprat using the proof assistant Coq. We have done a complete and proved development of a distributed reference counting algorithm. The main contribution of this work is the incremental construction starting from a simple abstraction. The distribution is gradually introduced over a series of refinement steps, until the final algorithm is obtained. This incremental construction allows us to validate step by step the real algorithm and to simplify its correctness proof. This work was presented in [15].

6.3. Formal methods for automation-systems engineering

Keywords: *fault tolerance*, *formal methods*, *industrial automation*, *safety assessment*, *system integrity*, *systems engineering*.

Participant: Dominique Méry.

Assessing system properties such as safety-integrity and fault-tolerance is a challenge for systems engineering since industrial automation is nowadays embedding intensive on-site and remote infotronics components. In joint work with G. Morel and J.-F. Pétin from CRAN (Centre de Recherche en Automatique de Nancy), and with J.-B. Léger from the consulting firm Prédict, we argue that proof-oriented formal techniques may bridge efficiently the gap with dependability-related practical techniques in order to mathematically check *a priori* the proof of fail-safety. Rationales, experiments and open issues are addressed with respect to combining the formal B event-based modelling framework, using the B proof assistant, with a formalized fault-tolerance modelling framework.

6.4. Generation of SystemC implementations from Event-B Models

Keywords: *B method*, *SystemC*, *hardware description*, *refinement*, *scheduling*.

Participants: Dominique Cansell, Dominique Méry, Cyril Proch.

Inspired from the results of the EQUAST project (see section 7.1), we propose a more general design method for proof-supported hardware-software codesign. The starting point of the method are models written in Event-B; these models help a designer in his architectural choices by structuring tasks of modelled system, making use of the structure of the invariant properties and of refinement. We define a translation that produces SystemC code from B models; the SystemC language is used by designers of electronic chips to describe different components of a system. The generated code conserves the scheduling properties of source models and permits electronic tests to verify time or consumption constraints. The correctness of the translation is proved via a formalization of the semantics of the SystemC scheduler [16]. A specific SystemC program can be represented with a B event-based model. This model is obtained by instantiating the generic model describing SystemC with the model of the concrete system. The approach has been presented in [10], [8], [17] and has been applied successfully for the DVB chip developed in the EQUAST project.

6.5. Translation from BHDL to ACL2

Keywords: *ACL2*, *B method*, *hardware description*, *refinement*, *theorem proving*.

Participants: Dominique Cansell, Yann Zimmermann.

Vendors of components for embedded systems usually provide models in the form of a synthesizable VHDL descriptions; these are not proved correct with respect to a specification. We propose to use the language BHDL, a sub-language of B for the hardware implementation level, as a specification language for synthesizable models. A BHDL specification is obtained by refinement from a B specification of the system under design. Then we propose to prove that the provided VHDL model is a correct refinement of the BHDL specification using the ACL2 theorem prover. This theorem prover was chosen because ACL2 has better performance for such low-level descriptions compared to the theorem prover of AtelierB; moreover there already existed a translator from VHDL to a model accepted as input of the ACL2 theorem prover. In collaboration with with Diana Toma-Moisic at the TIMA-IMAG Laboratory, we defined the translation from BHDL to ACL2 and the proof obligations that have to be discharged using ACL2 to ensure the correctness of the refinement.

Compared to the usual development process of the B method (top-down process), this defines a “meet in the middle” process, where the system is refined and some sub-modules are not refined to the implementation level with B but proved to be correctly implemented by an existing VHDL model.

This work was presented at the conference ZB 2005 [25].

6.6. Transaction Level Modeling (TLM)

Keywords: *ACL2, B method, hardware description, refinement, theorem proving.*

Participants: Dominique Cansell, Yann Zimmermann.

Electronic systems may include hardware modules as well as software modules (so-called systems on chip). To improve the development time, designers have defined a “transaction level” of SystemC (called SystemC-TLM), standardized this year, where communication between modules is done in a more abstract way than at the implementation level (and modules themselves may be described more abstractly). Such a transaction level model is faster to simulate and allows the software to be tested on abstract platforms. In this way, testing can be simplified, and the overall development process can be faster since testing and hardware development can occur in parallel. A problem is to ensure that the final synthesizable model of the hardware corresponds to the transaction level given to the software development team. We believe that the B method is a good method to ensure that there is a refinement relation between the translation level and the synthesizable level.

We have defined a BTLM language that is the B representation of the TLM level of SystemC and rules to translate BTLM model to SystemC-TLM.

6.7. Combining SMT Solvers and Interactive Proof Assistants

Keywords: *SMT solvers, decision procedures, proof assistants, theorem proving.*

Participants: Pascal Fontaine, Kamal Kant Gupta, Stephan Merz, Leonor Prensa Nieto, Alwen Tiu.

Formal system verification calls for expressive specification languages, but also requires highly automated tools. These two goals are not easy to reconcile, especially if one also aims at high assurances for correctness. Interactive proof assistants such as Coq or Isabelle provide rich logics such as higher-order logic, expressive type theories or Zermelo-Fränkel set theory. Moreover, their implementation is based on a small trusted kernel, which can be validated by inspection, decreasing the probability of errors. However, these logics are undecidable, and automation is limited. We investigate techniques to combine the interactive proof assistant Isabelle/HOL with decision procedures for fragments of first-order theories, with the aim of improving the automation possible for system verification. Specifically, we are interested in SMT (Satisfiability Modulo Theories) solvers. These tools group various decision procedures around a propositional SAT solver. In a nutshell, the SAT solver produces models for a Boolean abstraction of the input formula, and the decision procedures attempt to successively eliminate these models, further constraining the Boolean abstraction. The input formula is unsatisfiable if and only if (after some number of iterations) the Boolean abstraction has no more models.

In a first step, we have written translations from a fragment of Isabelle/HOL to the input language (SMT-LIB) of SMT solvers, allowing us to call these tools as oracles (trusted external reasoners) from within Isabelle/HOL. The problem with this approach lies with the translation from a polymorphic, higher-order input language to a (multi-sorted) first-order language, which is prone to errors, resulting in theorems with lower guarantees for soundness than for theorems that rely solely on the Isabelle kernel.

We have therefore studied ways to make SMT solvers produce proofs, rather than just a yes/no answer, and to have these proofs certified by the kernel of Isabelle/HOL, leading to theorems with the standard soundness guarantees provided by Isabelle. This approach has been first implemented for the core SAT solver (for both MiniSAT and zchaff), and it is now part of the Isabelle 2005 standard distribution. We have since extended the approach for the theory of quantifier-free first-order logic with uninterpreted function and predicate symbols. We have tested our approach on a number of examples (see also section 6.8) and have found that we can prove significantly more theorems than the existing automatic tactics in Isabelle. Ongoing work concerns the extension of this approach to further theories, such as linear arithmetic and fragments of set theory. Also, the efficiency should be further improved; it is mainly limited by inefficient data structures of the Isabelle kernel.

Preliminary results have been published in [19]; a longer paper has been accepted for publication at TACAS 2006.

6.8. Verification of Clock Synchronization Algorithms

Keywords: *clock synchronization, decision procedures, theorem proving*.

Participants: Damian Barsotti, Leonor Prensa Nieto, Alwen Tiu.

In relation with the work on combining the proof assistant Isabelle with decision procedures for first-order theories for the verification of distributed algorithms and systems, we have studied a family of clock synchronization algorithms. We first formalize in Isabelle/HOL an abstract framework for such algorithms proposed by Schneider [40]. Then we verify that the convergence functions used in two clock synchronization algorithms, namely, the Interactive Convergence Algorithm (ICA) of Lamport and Melliar-Smith [38] and the Fault-tolerant Midpoint algorithm of Lundelius-Lynch [41], satisfy Schneider's general conditions for correctness. The proofs are completely formalized in Isabelle/HOL. We identify the parts of the proofs which are not fully automatically proven by Isabelle built-in tactics and show that many of these proofs can be handled by automatic first-order provers with support for arithmetic like ICS and CVC Lite.

The formalization of Schneider's framework has been accepted at the Archive of Formal Proofs [29], and a paper on our results and experiences has been published at AVoCS 2005 [14].

6.9. Formal Verification of DiskPaxos

Keywords: *Isabelle/HOL, Paxos, consensus algorithm, theorem proving*.

Participants: Mauro Jaskelioff, Stephan Merz.

As another case study for investigating opportunities for the combination of different verification tools, we have formally verified the correctness of the DiskPaxos algorithm [34] due to Gafni and Lamport for the distributed consensus problem. Lamport's paper gives a TLA$^+$ model and outlines a rigorous (but informal) proof of its correctness. We have encoded Lamport's model in Isabelle/HOL and have fully elaborated the proof using the proof methods available in Isabelle/HOL. The development has been accepted at the Archive of Formal Proofs [28].

The formal verification certifies the correctness of Lamport's arguments. As expected, we discovered a few omissions in both the formulation of the invariant and in the correctness proof. In particular, we noticed an omission that has not been detected during a previous (incomplete) verification of DiskPaxos by Pacheco using ACL/2. He overlooked the problem because sets are encoded as inductive data types in ACL/2, implying that all sets are finite. The degree of automation of our proof could be raised by providing more support for reasoning about sets (see sections 6.7 and 5.3).

6.10. Combining Lists with Non-Stably Infinite Theories

Keywords: *combination of theories, decision procedures, lists, theorem proving*.

Participant: Pascal Fontaine.

In program verification one has often to reason about lists over elements of a given nature. Thus, it becomes important to be able to combine the theory of lists with a generic theory T modeling the elements. This combination can be achieved using the Nelson-Oppen method, provided that T is stably infinite, i.e. if any set of literals satisfiable in T has an infinite model. This condition can be restrictive, in particular for finite data types.

In the work published in [20] we relax the stable-infiniteness requirement. More specifically, we provide a new method that is able to combine the theory of lists with any theory T of the elements, regardless of whether T is stably infinite or not. The crux of our combination method is to guess an arrangement over a set of variables that is larger than the one considered by Nelson and Oppen. Furthermore, our results entail that it is also possible to combine T with the more general theory of lists with a length function.

This provides a decision procedure for quantifier-free formulas that contain list operators (car, cdr, cons), linear arithmetic, the length function, and any theory (stably-infinite or not) to handle the list elements.

6.11. Encoding TLA$^+$ in Isabelle

Keywords: *Isabelle, TLA, theorem proving*.

Participant: Stephan Merz.

In cooperation with Leslie Lamport, we have started to encode TLA$^+$ as a new object logic in the interactive proof assistant Isabelle. This encoding will eventually eliminate the need for translation from TLA$^+$ to Isabelle/HOL or Isabelle/ZF, whose logical foundations are different. Much of this work has been carried out during a visit to the Silicon Valley laboratory of Microsoft Research in August. So far, the encoding provides first-order logic, elementary set theory, functions, and natural numbers. It instantiates the automated reasoning tools provided by Isabelle for TLA$^+$. Ongoing work concerns the extension to the full TLA$^+$ data language, and the optimization of the automated proof methods. Eventually, Isabelle/TLA$^+$ is intended to become the centerpiece of an integrated proof environment for TLA$^+$.

6.12. Predicate diagrams for real-time systems

Keywords: *predicate abstraction, real-time systems*.

Participants: Eun Young Kang, Stephan Merz.

We study extensions of predicate diagrams (see section 3.2) for real-time systems. The systems we study are described as extended time graphs (XTG), a notation that extends timed automata by a data description language. We have defined a variant of predicate diagrams appropriate for timed systems; in particular, by distinguishing between discrete and time-passing transitions. The DIXIT tool (see section 5.2) is used to analyze these diagrams in the standard way. The work has been presented in [23] and [22]. Ongoing work concerns the generation of timed predicate diagrams using abstraction refinement.

6.13. System co-development in B and UML

Keywords: *B method, UML, refinement*.

Participants: Houda Fekih, Arnaud Lanoix, Stephan Merz.

Formal and semi-formal methods of system development emphasize different aspects of system models: whereas semi-formal methods such as UML emphasize system structure, formal methods offer a precise semantics and enable correctness and performance analysis. Previous work on reconciling the two views of system modeling has concentrated on associating formal semantics with models written in semi-formal languages. Unfortunately, the resulting translations are either restricted to small fragments of UML or tend to produce unnatural formal models that are difficult to understand and to verify. We propose to consider formal and semi-formal models as two complementary views of a single underlying system that evolve in parallel. Relationships that exist between formal and semi-formal representations should be maintained across refinements, and revisions in one model should be reflected in the other. We have defined translations from B event systems to UML class diagrams and state machines, and have in particular identified a number of idioms that give rise to characteristic structures in the UML model. A paper describing these translations has been accepted for publication at the Software Verification track of the ACM Symposium on Applied Computing in Dijon in 2006. In joint work with the DEDALE team of LORIA, we have also started to study refinement notions within UML that correspond to the concepts familiar from formal methods such as B.

6.14. Specification of mobile systems

Participant: Stephan Merz.

In joint work with Martin Wirsing and Júlia Zappe from the University of Munich, we have proposed MTLA, an extension of Lamport's Temporal Logic of Actions [37] by spatial modalities for the specification and verification of systems based on mobile code, focusing on appropriate notions of refinement and their representation in the logic.

In 2005, the work on the meta-theory of MTLA has been continued; in particular, the satisfiability and model checking problems for propositional MTLA have been shown to be decidable, and a complete axiomatization has been obtained for this fragment. These results are presented in the doctoral thesis of Júlia Zappe at the University of Munich, defended in September 2005, and for which Stephan Merz was a reviewer. We have also applied MTLA to describe class diagrams and state machines of Mobile UML and to derive verification conditions. This work has been published recently in Theoretical Computer Science [9].

6.15. Model checking using alternating automata

Keywords: *Spin, alternating automata, model checking, temporal logic*.

Participant: Stephan Merz.

The automata-theoretic approach to LTL model checking has traditionally been based on a translation of LTL formulae to Büchi automata. Unfortunately, this translation is of exponential complexity, limiting the size of formulae that can effectively be verified in this way, and several heuristics have been developed to avoid the exponential blow-up as far as possible.

We have designed an alternative algorithm for LTL model checking based on linear weak alternating automata. The translation of LTL to this class of automata is of linear complexity, the exponential factor is delayed until the emptiness problem has to be decided, opening the way to verify large LTL formulas. (Such formulas may in particular arise during the verification of liveness properties for Boolean abstractions of models.) In joint work with Moritz Hammer from the University of Munich, the algorithm has been implemented within the Spin model checker, and it has been shown to significantly outperform classical model checking for large formulas. The work has been published at TACAS 2005 [21].

6.16. Formally Verifying Information Flow Type Systems for Concurrent and Thread Systems

Keywords: *Isabelle/HOL, confidentiality, non-interference, theorem proving, type systems*.

Participant: Leonor Prensa Nieto.

Information flow type systems provide an elegant means to enforce confidentiality of programs. In collaboration with Gilles Barthe from INRIA Sophia-Antipolis and using the proof assistant Isabelle/HOL, we have specified an information flow type system for a concurrent language featuring primitives for scheduling, and shown that typable programs are non-interfering for a possibilistic notion of non-interference. Our language and type system generalize previous work of Boudol and Castellani [33]. The development constitutes to our best knowledge the first machine-checked account of non-interference for a concurrent language. This work has been published in [39].

We have further extended the language, in particular by including arrays and removing several simplifying but unnecessary restrictions in the syntax and type system of [33]. Also, the mechanical formalization has been modified. We use locales to encapsulate the different layers of the language. We abstract from the particular implementation of the sequential and parallel sublanguages and identify for each sublanguage the necessary conditions (as assumptions in the corresponding locale) that ensure non-interference clarifying the relationship among the different layers. The proof of non-interference for the type system of the concrete implementation of the language of [39] is then proven simply by instantiating the sublanguages and proving the validity of the required assumptions.

We illustrate the generality of our language and the usefulness of our type system with a medium size example.

This work has been accepted for publication in a special issue of the Journal in Computer Security (JCS).

6.17. Modeling of automated systems in B and TLA$^+$

Keywords: *B method, TLA, automated systems, proof.*

Participants: Jacques Jaray, Olfa Mosbahi.

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 automated systems. Olfa Mosbahi visited LORIA from June to September. The work has resulted in a paper accepted in the International Conference on Computer Systems and Applications 2006.

6.18. Type inference for TLA$^+$ specifications

Keywords: *TLA, type inference.*

Participants: Loïc Fejoz, Stephan Merz.

The specification language TLA$^+$ is based on (untyped) Zermelo-Fränkel set theory, which gives users much freedom for expressing mathematical constructions. For system specifications, a stronger typing discipline is often useful. With current TLA$^+$ tools, type errors are usually detected during model checking. In his master's thesis [27], Loïc Fejoz investigates techniques for inferring types of TLA$^+$ expressions and state variables in actual system specifications. This has led to the definition of a type system for set theory with union and function types. We have also given rules for type checking and type inference, which have been implemented in a prototype and experimented on a number of specifications of various complexity. We have been able to quickly infer type approximations that have helped to discover problems in modules that were not amenable to model checking, but we believe that this work should be complemented by the static verification of user-written typing annotations.

6.19. Work on METAPOST, its extensions and 3D-prototyping

Keywords: *METAPOST, graphics, layout.*

Participant: Denis Roegel.

Denis Roegel has continued to develop extensions to METAPOST addressing various abstract representations of objects. In particular, he is interested in means to ease the prototyping of 3-dimensional objects by using the METAPOST interface, while making use of state of the art rendering environments such as OpenGL. A first result of this work was presented at the EuroTeX conference[24].

Further work will address the use of OpenGL NURBS surfaces from a METAPOST interface.

We have also studied the METAPOST implementation of David Eppstein's algorithm for Apollonius' circle problem[11].

Finally, Denis Roegel is currently contributing as a coauthor to the 2nd edition of the *LaTeX Graphics Companion* book, due in 2006.

7. Contracts and Grants with Industry

7.1. RNRT Equast

Keywords: *digital video broadcast, fault hierarchy, modelling, quality of service.*

Participants: Dominique Cansell, Dominique Méry, Cyril Proch.

The RNRT Equast project was initiated in November 2002 and was finished successfully in April 2005. Within this project, we cooperated with the following partners: Thalès B&M, TDF, and LIEN (Laboratoire d'Instrumentation d'Electronique de Nancy). Its objective was to develop a circuit to analyze the image quality of terrestrial digital TV as specified by a set of 30 parameters defined in the TR 101 290 standard of the ETSI (European Telecommunications Standard Institute). The contribution of MOSEL was to provide an incremental and proved model of the system.

We have modelled the entire set of parameters in a series of seven refinements that incrementally introduce the parameters. This work allowed us

- to exhibit a hierarchy among the parameters defined in the standard and thus to justify the parameters determining the quality of service that had been introduced by TDF (initiated during the European projects QUOVADIS and MOSQUITO),
- to validate all parameters with respect to the requirements in discussions with our partners,
- to derive from our model suggestions for global and reconfigurable architectures of the FPGA to be developed by our colleagues from the LIEN, and
- to derive from our model SystemC code, making use of the "sensitivity" mechanism of SystemC modules.

To validate our translation from B model to SystemC code we have modeled the scheduler of SystemC in event B. The semantics of SystemC is based on its scheduling algorithm described in the language reference manual and we develop a B model of the scheduling. The B *scheduling* model leaves unspecified parameters depending on the simulated SystemC program and those parameters are instantiated from the operational semantics of the developed SystemC program. By instantiation, we obtain a B abstract model of the simulated program and we can study properties of the SystemC program by simulation. B models are completely validated by the proof assistant of the event B method. Finally, our models provide a sound framework for understanding the scheduling process.

Our results on the EQUAST project have been published in journals [10], [8], at the IBC 2005 conference [12], and the ISOLA 2005 workshop [16].

8. Other Grants and Activities

8.1. Regional cooperation: QSL platform and operation DIXIT'

Keywords: *abstract interpretation, abstraction, predicate diagrams, refinement.*

Participants: Loïc Fejoz, Pascal Fontaine, Dominique Méry, Stephan Merz, Leonor Prensa Nieto, Alwen Tiu.

We are participating in the QSL (*Qualité et Sûreté des Logiciels*) theme of the research hub "Intelligence Logicielle" at the LORIA laboratory. Members of our group are actively participating in the construction of the "QSL platform" that aims at the combination of deductive tools for system verification (see also sections 6.7, 6.8, and 6.9).

The operation DIXIT' was approved by the QSL council in the spring of 2003 and it terminated successfully in 2005. Its objective was research into the representation of Boolean abstractions of distributed and reactive systems as predicate diagrams and the implementation of an integrated tool set for their manipulation (see also section 5.2).

8.2. National cooperation

Our team is participating in two national research projects concerning the security of computer-based systems (ACI Sécurité Informatique) named CORSS and DESIRS.

8.2.1. 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 project (Composition et raffinement de systèmes sûrs) includes participants with a background in system development and those 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.

8.2.2. Project DESIRS

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

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

The DESIRS project (Développement de systèmes informatiques par raffinement des contraintes sécuritaires) studies 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). This year we studied how the refinement concepts of the B method can be extended to preserve certain possibility properties, in view of accomodating access control policies described using the OrBAC model [35]. Nazim Benaïssa prepared his master's thesis [26] under the supervision of Dominique Méry on the design of systems using Event-B based on OrBAC models.

8.3. International cooperations

8.3.1. Projects NSF CNRS

Participants: Dominique Cansell, Dominique Méry, Stephan Merz.

Since 1998, MOSEL is actively cooperating with the team led by Prof. Ganesh Gopalakrishnan at the University of Utah. Dominique Méry and Stephan Merz visited the group in July for mutual presentation of ongoing work, in particular concerning the verification of cache coherence protocols.

A second cooperation that is funded by an agreement between the U.S. National Science Foundation (NSF) and CNRS concerns the team led by Prof. Beverly Sanders at the University of Florida at Gainesville.

8.3.2. Mediterranean, North Africa, and Middle East

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.

9. Dissemination

9.1. Program committees and conference organisation

- Dominique Cansell was a member of the program committees of ZB 2005 and AFADL 2006.
- Dominique Méry was a member of the program committees of IFM 2005 and ICFEM 2005.
- Stephan Merz served on the program committees of FASE 2005 and IFM 2005.

9.2. Tutorials, invited talks, panels

- Stephan Merz and Leonor Prensa Nieto gave courses on TLA$^+$ and on theorem proving using Isabelle/HOL at a summer school in Rio Cuarto, Argentina, in February.
- Stephan Merz was invited to participate in the Beyond-the-Horizon workshop on Software-Intensive Systems at the University of Koblenz in September.
- Pascal Fontaine gave a talk at the Nancy-Saarbrücken workshop on Logics, Proofs, and Programs in October in Nancy.

9.3. Theses, habilitations, academic duties

- Dominique Cansell is representing the B group of the GDR ALP at Nancy. He was responsible within MOSEL for the EQUAST project (see section 7.1).
- Dominique Cansell is a member of the hiring committee 27 at the University of Metz.
- Dominique Cansell wrote a report on the Ph.D. thesis of J.-M. Mota (Université Évry Val d'Essonne).
- Jacques Jaray is director of studies at the École des Mines of Nancy.
- Dominique Méry wrote reports on the Docent thesis of Marina Walden at Abo Akademi, Turku, Finland, and on the Ph.D. thesis of Syrine Ayadi at Université de Tunis El-Manar.
- Dominique Méry is a member of the scientific council of the University Henri Poincaré Nancy 1.
- Dominique Méry is the director of the Master's program of computer science in Nancy.
- Dominique Méry is a member of the scientific council of the LORIA laboratory.
- Dominique Méry is a member of the IFIP Working Group 1.3 *Foundations of System Specification*.
- Dominique Méry is an expert for the French Ministery of Education (DS9).
- Dominique Méry is the scientific leader of the DIXIT' operation (see section 8.1).
- Stephan Merz wrote a report for the Ph.D. thesis of Júlia Zappe (Ludwig-Maximilians University, Munich).
- 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 directorate 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.
- Since the fall of 2005, 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 2005.

- Dominique Cansell taught a course on the B method at the Master's program at the University of Poitiers.
- Dominique Cansell and Dominique Méry gave a course in the Master's program at Nancy on the specification and modelling of computer-based systems.
- Dominique Méry gives courses on the specification of computer-based systems at the Ecole Supérieure d'Electricité of Metz. He also teaches on the quality and safety of programs in DESS courses. With Edufrance, he has instituted the exchange of foreign students at various levels of university education. He is director of international relations at ESIAL Nancy.
- Together with Olivier Bournez, Stephan Merz gave courses on algorithmic verification and on the semantics of parallel and distributed systems at the Master's program in Nancy.
- Denis Roegel taught a graduate course on scientific typesetting for prospective PhD students from the Nancy 1 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.

Articles in refereed journals and book chapters

[7] J.-R. ABRIAL, D. CANSELL. *Formal Construction of a Non-blocking Concurrent Queue Algorithm (a Case Study in Atomicity)*, in "Journal of Universal Computer Science", vol. 11, n^o 5, 2005, p. 744–770, http://hal.inria.fr/inria-00000120/en/.

[8] D. Cansell, D. Méry, C. Proch. *Un système d'analyse de la qualité: de la norme au produit en passant par le raffinement*, in "Génie logiciel", n^{o} 73, 2005, p. 44–50, http://hal.inria.fr/inria-00000196/en/.

[9] A. Knapp, S. Merz, M. Wirsing, J. Zappe. *Specification and Refinement of Mobile Systems in MTLA and Mobile UML*, in "Theoretical Computer Science", December 2005, http://hal.inria.fr/inria-00000754/en/.

[10] D. Méry, D. Cansell, C. Proch, D. Abraham, P. Ditsch. *The challenge of QoS for digital television services*, in "EBU Technical Review", April 2005.

[11] D. Roegel. *Kissing Circles: A French Romance in METAPOST*, in "TUGboat", vol. 26, n^{o} 1, 2005, p. 10–17, http://hal.inria.fr/inria-00000659/en/.

Publications in Conferences and Workshops

[12] D. Abraham, D. Cansell, P. Ditsch, D. Méry, C. Proch. *Synthesis of the QoS for digital TV services*, in "First International Workshop on Incentive Based Computing - IBC'05, Amsterdam, The Netherlands", 2005, http://hal.inria.fr/inria-00000565/en/.

[13] J.-R. Abrial, D. Cansell, D. Méry. *Refinement and Reachability in Event_B*, in "ZB 2005: Formal Specification and Development in Z and B, Guilford, UK", H. Treharne, S. King, M. Henson (editors). , Lecture Notes in Computer Science, vol. 3455, Springer Verlag, April 2005, p. 222–241.

[14] D. Barsotti, L. Prensa Nieto, A. Tiu. *Verification of Clock Synchronization Algorithms: Experiments on a combination of deductive tools*, in "Fifth International Workshop on Automated Verification of Critical Systems 2005 - AVoCS '05, Coventry, U.K.", 2005, http://hal.inria.fr/inria-00000638/en/.

[15] D. Cansell, D. Méry. *Formal and Incremental Construction of a Distributed Reference Counting Algorithm*, in "APPSEM 2005", M. Hoffmann, H.-W. Loidl (editors). , September 2005.

[16] D. Cansell, D. Méry, C. Proch. *Modelling SystemC scheduler by refinement*, in "IEEE ISoLA Workshop on Leveraging Applications of Formal Methods, Verification, and Validation - ISOLA'05, Columbia, U.S.A.", 2005, http://hal.inria.fr/inria-00000564/en/.

[17] D. Cansell, D. Méry, C. Proch. *Proved Design of Hardware Architecture*, in "Formal Specification and Development in Z and B (ZB 2005). Poster session, Guildford, U.K.", April 2005.

[18] L. Fejoz, D. Méry, S. Merz. *DIXIT: a Graphical Toolkit for Predicate Abstractions*, in "Fourth International Workshop on Automated Verification of Infinite-State Systems (AVIS 2005), Edinburgh, U.K.", R. Bharadwaj, S. Mukhopadhyay (editors). , 2005, p. 39–48, http://hal.inria.fr/inria-00000767/en/.

[19] P. Fontaine, K. K. Gupta, J.-Y. Marion, S. Merz, L. P. Nieto, A. Tiu. *Towards a Combination of Heterogeneous Deductive Tools for System Verification*, in "APPSEM 2005", M. Hoffmann, H.-W. Loidl (editors). , September 2005.

[20] P. Fontaine, S. Ranise, C. Zarba. *Combining Lists with Non-Stably Infinite Theories*, in "11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'04), Montevideo, Uruguay", F. Baader, A. Voronkov (editors). , Lecture Notes in Computer Science, vol. 3452, Springer-

Verlag, 2005, p. 51–66, http://hal.inria.fr/inria-00000481/en/.

[21] M. HAMMER, A. KNAPP, S. MERZ. *Truly On-The-Fly LTL Model Checking*, in "Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2005), Edinburgh, U.K.", N. HALBWACHS, L. D. ZUCK (editors). , Lecture Notes in Computer Science, vol. 3440, Springer-Verlag, 2005, p. 191–205, http://hal.inria.fr/inria-00000753/en/.

[22] E. KANG. *Real-Time system verification techniques based on abstraction/deduction and model checking*, in "Doctoral Symposium of the Fifth International Conference on Integrated Formal Methods - IFM'2005. Technical Report of MCS, TU-Eindhoven, Eindhoven, The Netherlands", J. ROMIJN (editor). , 2005, http://hal.inria.fr/inria-00000641/en/.

[23] E. KANG, S. MERZ. *Predicate Diagrams for the Verification of Real-Time Systems*, in "The Fifth International Workshop on Automated Verification of Critical Systems 2005 - AVoCS'05, Coventry, U.K.", R. LAZIC, R. NAGARAJAN, N. PAPANIKOLAOU (editors). , Elsevier, 2005, http://hal.inria.fr/inria-00000631/en/.

[24] D. ROEGEL. *MP2GL: prototyping 3D objects with METAPOST and OpenGL*, in "15e rencontre annuelle des utilisateurs européens de TeX - EuroTeX 2005, Pont-à-Mousson, France", March 2005, http://www.loria.fr/~roegel/TeX/eurotex2005roegel.pdf.

[25] Y. ZIMMERMANN, D. TOMA. *Component Reuse in B Using ACL2*, in "ZB 2005: Formal Specification and Development in Z and B: 4th International Conference of B and Z Users, Guildford, U.K.", H. TREHARNE, S. KING, M. HENSON, S. SCHNEIDER (editors). , Lecture Notes in Computer Science, vol. 3455, Springer-Verlag, 2005, p. 280–299, http://hal.inria.fr/inria-00000755/en/.

Miscellaneous

[26] N. BENAÏSSA. *Conception de systèmes avec le B événementiel à partir de modèles ORBAC*, Technical report, DEA Informatique, Nancy, 2005.

[27] L. FEJOZ. *Typage de TLA$^+$*, Technical report, DEA Informatique, Nancy, 2005.

[28] M. JASKELIOFF, S. MERZ. *Proving the Correctness of DiskPaxos*, June 2005, http://afp.sourceforge.net/entries/DiskPaxos.shtml, Archive of Formal Proofs.

[29] A. TIU. *Formalization of a Generalized Protocol for Clock Synchronization*, June 2005, http://afp.sourceforge.net/entries/GenClock.shtml, Archive of Formal Proofs.

Bibliography in notes

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

[31] 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.

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

[33] G. BOUDOL, I. CASTELLANI. *Noninterference for concurrent programs and thread systems*, in "Theoretical Computer Science", Special issue “Merci, Maurice. A mosaic in honour of Maurice Nivat”, vol. 281, n^o 1, 2002, p. 109–130.

[34] E. GAFNI, L. LAMPORT. *Disk Paxos*, in "International Symposium on Distributed Computing", 2000, p. 330–344.

[35] A. A. E. KALAM, R. E. BAIDA, P. BALBIANI, S. BENFERHAT, F. CUPPENS, Y. DESWARTE, A. MIÈGE, C. SAUREL, G. TROUESSIN. *Organization-Based Access Control*, in "4th Intl. Workshop Policies for Dist. Systems and Networks (Policy 2003), Lake Como, Italy", J. MOFFETT, F. GARCIA (editors). , IEEE Press, June 2003.

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

[37] L. LAMPORT. *The Temporal Logic of Actions*, in "ACM Transactions on Programming Languages and Systems", vol. 16, n^o 3, May 1994, p. 872–923.

[38] L. LAMPORT, P. M. MELLIAR-SMITH. *Synchronizing clocks in the presence of faults*, in "J. ACM", vol. 32, n^o 1, 1985, p. 52–78.

[39] L. PRENSA NIETO, G. BARTHE. *Formally Verifying Information Flow Type Systems for Concurrent and Thread Systems*, in "2nd ACM Workshop on Formal Methods in Security Engineering: From Specifications to Code (FMSE’04), Washington D.C., U.S.A.", M. BACKES, D. BASIN, M. WAIDNER (editors). , ACM SIGSAC, ACM, October 2004, p. 13–22.

[40] F. B. SCHNEIDER. *Understanding protocols for Byzantine clock synchronization*, Technical report, n^o 87-859, Department of Computer Science, Cornell University, August 1987.

[41] J. L. WELCH, N. LYNCH. *A New Fault-Tolerant Algorithm for Clock Synchronization*, in "Information and Computation", vol. 77, 1988, p. 1-36.