0.1 WP 5: Visits between sites

There is plenty of interaction between the sites, most of them is shown in the table on page 1.

Table 1: List of visits between sites:

From	To	Who
Paris-Sud	Nottingham	Nicolas Oury
Bergen	Paris-Sud	Marc Bezem
INRIA Sophia	Paris-Sud	Philippe Audebaud
Nijmegen	Paris-Sud	Pierre Corbineau

The following is a short list of talks given during some of the short visits:

- Marc Bezem from Bergen visited Paris-Sud 30 Sept 2005. He gave a talk about " Mechanizing projective geometry using Coherent Logic". [0]

  Coherent logic (CL) is a fragment of FOL extending resolution logic in that it allows certain existential quantifications. CL has a natural proof theory, reasoning in CL is constructive and proof objects can easily be obtained. A substantial number of reasoning problems (e.g., in confluence theory, lattice theory and projective geometry)can be formulated directly in CL without any clausification or Skolemization. This gives some additional benefits in terms of guiding an automated theorem prover, efficiency of the proof search and reusing the proof objects in other logical frameworks. After a short introduction to CL, he discussed a number of examples in projective geometry which have been formalized in Coq.

- Philippe Audebaud from INRIA Sophia-Antipolis visited Paris-Sud and gave talk about the semantics of the probabilistic language Λ_O.

- Pierre Corbineau from Radboud university at Nijmegen visited Paris-Sud on 30 June 2006 and gave a talk on a declarative proof language for Coq.

- Because of geographical proximity, there are strong interactions between the sites Paris-Sud, INRIA Futurs and Université Paris 7. There is a common seminar between Paris-Sud and INRIA Futurs. J-C Filliâtre, J. Signoles and N. Oury gave seminars in Paris 7.

1 Scientific collaboration inside the Types project

We promote exchange of students. P. Corbineau got his PhD in Paris-Sud in September 2005 and started a post-doc position at Rabdoub university in Nijmegen. Nicolas Oury from Paris Sud will defend his PhD in September 2006 and got a Marie-Curie fellowship for a post-doc at university of Nottingham.

2 Industrial cooperation

We are collaborating with Dassault Aviation and France Telecom R&D in the area of proofs of C programs. We also have a collaboration with the Axalto

company (a smart-cards manufacturer) on proofs of Java and C programs, Java card applets and operating systems. J. Andronick defended her PhD-thesis at Paris Sud in march 2006, the work was done part-time in Axalto. Th. Hubert (Dassault), N. Rousset (Axalto), Y. Moy (France telecom R&D) are studying for their PhD part-time in the industry and part-time in our laboratory.

There was a collaboration between Orsay, Grenoble and the industrial subsite France Telecom R&D in the AVERROES national project (analysis and verification for the reliability of embedded systems) which finished in march 2006 (`http://www-verimag.imag.fr/AVERROES`).

Paris-Sud is participating to the french competitiveness cluster System@tic (see `http://www.systematic-paris-region.org`). In this cluster, the main industrial and academic research centers in the Ile-de-France Region are collaborating in the area of complex systems.

3 Coauthored papers and presentations

Refereed conference papers

- Philippe Audebaud and Christine Paulin-Mohring. Proofs of randomized algorithms in coq. In Tarmo Uustalu, editor, *Mathematics of Program Construction, MPC 2006*, volume 4014 of *Lecture Notes in Computer Science*, Kuressaare, Estonia, July 2006. Springer-Verlag

Software

- The Coq Development Team. *The Coq Proof Assistant Reference Manual – Version V8.1*, July 2006. `http://coq.inria.fr`

3.1 Paris-Sud/Grenoble/France Telecom R&D

3.1.1 Correctness of Computer Systems

Proving C or Java programs Our main activity is related to program verification. We mainly focus on the verification of behavioral specifications for programming languages such as C, Java and ML. We develop a tool "Why" (see `http://why.lri.fr`) which is a verification conditions generator: from an annotated program written in a small imperative language with Hoare logic-like specification, it generates conditions expressing the correctness and termination of the program. These verification conditions can be generated for several existing provers, including interactive proof assistants (Coq, PVS, HOL Light, Mizar) and automatic provers (Simplify, haRVey, CVC Lite).

On top of this tool, we built a system called Krakatoa (`http://krakatoa.lri.fr`) which verifies Java source code annotated with the Java Modeling Language (JML). The main challenge was the design of a suitable model for the Java memory heap in order to tackle programs with possible aliases [15]. This tool has been adapted by C. Marché and N. Rousset in order to handle JavaCard transaction mechanism [14].

J.-C. Filliâtre and C. Marché designed a similar tool called Caduceus for dealing with C programs. This tool is used by the France Telecom R&D subsite in order to analyse use of memory. Y. Moy supervised by P. Crégut and C. Marché is designing a method to automatically generate specifications (loop invariant, precondition) corresponding to an appropriate use of pointers.

J. Andronick worked in the Axalto company on an adaptation of the Caduceus tool in order to specify and prove embedded source code on smart cards, possibly using union types and casts between structures and arrays [1].

Reasoning on functional programs J. Signoles [21] supervised by J-C. Filliâtre proposed an extension of mini-ML with types seen as ordinary expressions and expressions interpreted as specifications. Using a single extra construction (demonic application), he is able to express powerful specifications and a refinement relation in order to develop correct programs.

M. Sozeau supervised by C. Paulin, designed a language with a subset type (in the spirit of the PVS language) which is convenient for programming with (a restricted class of) dependent types. He proposed a translation of a term in this language to a Coq term containing existential variables corresponding to type-checking conditions. He developed a formal proof of correctness of the translation and designed a prototype implementation available in CoqV8.1 [22].

Reasoning on randomized programs C. Paulin together with Ph. Audebaud (from INRIA Sophia-Antipolis) proposed a method for representing randomized algorithms in Coq using a monadic interpretation translating randomized expressions into distributions [2]. A library has been designed in Coq for representing the interval $[0, 1]$, probabilistic distributions and randomized algorithms [20].

Applications of proof assistant to security Coq was used in the framework of security for bank applications. The API of IBM's Common Cryptographic Architecture used in most ATMs, was known to be flawed: secrets can

be disclosed using regular function calls and properties of the bitwise exclusive or operator. Courant and Monin showed that this cannot happen in a suitable modification of the API. The proof was designed in Coq: the proof tool played an essential role in the discover of the right invariants [9, 8].

Floating-point arithmetic Numeric computations use floating-point numbers to approximate exact arithmetic. Unfortunately, this use can falsify a program correct on real numbers. The use of a proof assistant is especially useful as some computations, like polynomial evaluation [5], are commonplace and as floating-point arithmetic may have unexpected behaviors [3].

3.1.2 Foundational Research

In the course of the proof of the security API, a new methodology was experimented for expressing a non-trivial function, namely the normalization of terms quotiented by the algebraic properties of xor. This methodology is based on a stacking of dependent types, instead of general results on AC-rewriting relations [17].

Automatic deduction Integrating automatic deduction into type theory is a long term research.

S. Conchon designed an automatic proof procedure for first-order logic with equality and arithmetic in order to solve proof obligations generated by checking correctness of programs. His implementation is functional with the goal to integrate it in proof assistants using certification or trace generation.

S. Lescuyer supervised by E. Contejean and S. Conchon designed a method for translating a problem defined in a multi-sorted polymorphic theory into a formula adapted to automatic provers based on mono-sorted logic.

E. Contejean continue her long-term project of cross-fertilizing rewriting techniques and type theory. She is designing a large Coq library (described at `http://www.lri.fr/~contejea/COQ/doc/`) on rewriting, using an efficient representation of terms, similar to the one used in rewriting tools like CiMe. She developed a proof of termination for the RPO order and is currently working on unification.

Pattern matching with dependent types N. Oury supervised by C. Paulin designed a method for analysing pattern-matching completeness with dependent types using techniques derived from abstract interpretation [19].

3.1.3 Formal Mathematics and Mathematics Education

Ideas coming from the TYPES community were extensively used for teaching the basis of logic reasoning in a new course for first year undergraduate (L1) students of the university of Grenoble (UJF). The emphasis is put on deduction rules instead of truth tables. Attendees: 300 students in computer science, mathematics, biology, chemistry and physics. Lecturer: JF Monin.

3.1.4 Proof technology

Hermes `http://www-verimag.imag.fr/~Liana.Bozga/home/hermes.html`, is a tool dedicated to the automatic verification of secrecy properties in cryptographic protocols, has been connected to Coq in order to certify the positive results it may produce as output. Basically, Hermes computes a fixed point in an abstract representation of an infinite state system, while providing a Coq proof script which is is verified off-line by Coq [11].

We collaborated with the LogiCal team (INRIA Futurs) by extending the parameter condition of inductive definitions in the new version V8.1 of the Coq Proof Assistant [23].

Publications

Refereed journal papers

- Salvador Lucas, Claude Marché, and José Meseguer. Operational termination of conditional term rewriting systems. *Information Processing Letters*, 95:446–453, 2005
- Jean-François Monin and Philippe Chavin. Coq. In H. Habrias and M. Frappier, editors, *Software Specification Methods, An Overview Using a Case Study*, ISTE, chapter 16. Hermès Science, April 2006

Refereed conference papers

- Philippe Audebaud and Christine Paulin-Mohring. Proofs of randomized algorithms in coq. In Tarmo Uustalu, editor, *Mathematics of Program Construction, MPC 2006*, volume 4014 of *Lecture Notes in Computer Science*, Kuressaare, Estonia, July 2006. Springer-Verlag
- Sylvie Boldo. Pitfalls of a full floating-point proof: example on the formal proof of the veltkamp/dekker algorithms. In *Proceedings of the third International Joint Conference on Automated Reasoning (IJCAR)*, Seattle, USA, August 2006
- Sylvie Boldo and César Muñoz. Provably faithful evaluation of polynomials. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing*, volume 2, pages 1328–1332, Dijon, France, April 2006
- Sylvain Conchon and Jean-Christophe Filliâtre. Type-Safe Modular Hash-Consing. In *ACM SIGPLAN Workshop on ML*, Portland, Oregon, September 2006
- Judicaël Courant and Jean-François Monin. Defending the bank with a proof assistant. In *6th International Workshop on Issues in the Theory of Security (WITS '06)*, Vienna, March 2006
- Judicaël Courant and Jean-François Monin. Faire garder la banque par un Coq. In *Actes des dix-septièmes journées francophones des langages applicatifs*, pages 25 – 39, January 2006
- Jean-Christophe Filliâtre. Backtracking iterators. In *ACM SIGPLAN Workshop on ML*, Portland, Oregon, September 2006

- Romain Janvier, Yassine Lakhnech, and Michaël Périn. Certification of Cryptographic Protocols by Abstract Model-Checking and Proof Concretization. In *ITCES'2006*, San Jose, April 2006
- Claude Marché and Christine Paulin-Mohring. Reasoning about Java programs with aliasing and frame conditions. In J. Hurd and T. Melham, editors, *18th International Conference on Theorem Proving in Higher Order Logics*, Lecture Notes in Computer Science. Springer-Verlag, August 2005
- Claude Marché and Nicolas Rousset. Verification of Java Card applets behavior with respect to transactions and card tears. In *4th IEEE International Conference on Software Engineering and Formal Methods (SEFM'06)*, Pune, India, September 2006
- Jean-François Monin and Judicaël Courant. Proving termination using dependent types: the case of xor-terms. In *Trends in Functional Programming 2006*, Nottingham, April 2006
- Nicolas Oury. Extensionality in the Calculus of Constructions. In J. Hurd and T. Melham, editors, *18th International Conference on Theorem Proving in Higher Order Logics*, Lecture Notes in Computer Science. Springer-Verlag, August 2005

Talks

- Sylvie Boldo. Veltkamp & Dekker revisited. TYPES Workshop on Numbers and Proofs, June 2006
- Nicolas Oury. Pattern matching coverage using approximations. TYPES 2006, long talk, April 2006
- Matthieu Sozeau. Subset coercions in Coq. TYPES 2006, long talk, April 2006
- Christine Paulin-Mohring. A library for reasoning on randomized algorithms in Coq. Description of a Coq contribution, Université Paris Sud, January 2006. AVERROES project

Dissertations

- June Andronick. *Modélisation et vérification formelles de systèmes embarqués dans les cartes à microprocessur. Plateforme Java Card et Système d'exploitation.* Thèse de doctorat, Université Paris-Sud, March 2006
- Pierre Corbineau. *Démonstration Automatique en Théorie des Types.* Thèse de doctorat, Université Paris-Sud, September 2005
- Claude Marché. *Preuves mécanisées de Propriétés de Programmes.* Thèse d'habilitation, Université Paris 11, December 2005
- Julien Signoles. *Extension de ML avec raffinement: syntaxe, semantiques et système de types.* Thèse de doctorat, Université Paris-Sud, July 2006