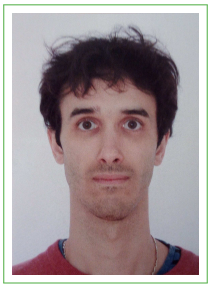
Vincent Rahli

Research Associate

6 Avenue de la Fonte
L-4364, Esch-sur-Alzette
☎ *(00352) 661 268 081*
✉ *vincent.rahli@gmail.com*
https://vrahli.github.io/



Interests

- Specification, implementation, and verification of fault-tolerant distributed systems.
  E.g. Paxos, PBFT, MinBFT
- Foundations of intuitionistic logic.
  E.g. bar induction, continuity, choice sequences
- Theory and meta-theory of proof assistants.
  E.g. Agda, Coq, Nuprl
- Functional programming language foundations, concepts and paradigms.
  E.g. λ-calculus, closures, continuations, side effects

Experience

2015–Present **Research Associate**, *Interdisciplinary Centre for Security, Reliability and Trust (SnT) University of Luxembourg*, Luxembourg.

2011–2015 **Research Associate**, *Department of Computer Science, Cornell University*, Ithaca, NY, USA.

Education

2006–2010 **PhD in Computer Science**, *Heriot-Watt University*, Edinburgh, UK, PhD advisors: Professor Fairouz Kamareddine and Doctor Joe B. Wells.
Title: "Investigations in intersection types: Confluence, and semantics of expansion in the λ-calculus, and a type error slicing method"

2005–2006 **Master's degree (MPRI—Parisian Master of Research in Computer Science)**, *University of Paris 7*, France, Computer Science.

Publications

Conferences

[1] **A Verified Theorem Prover Backend Supported by a Monotonic Library**, *Vincent Rahli, Liron Cohen, and Mark Bickford*, International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR), 2018.

[2] **Computability Beyond Church-Turing via Choice Sequences**, *Vincent Rahli, Liron Cohen, Mark Bickford, and Robert L. Constable (contribution order)*, Symposium on Logic in Computer Science (LICS), 2018.

[3] **Velisarios: Byzantine Fault-Tolerant Protocols Powered by Coq**, *Vincent Rahli, Ivana Vukotic, Marcus Völp, and Paulo Esteves-Verissimo*, European Symposium on Programming (ESOP), 2018.

[4] **Bar Induction: The Good, the Bad, and the Ugly**, *Vincent Rahli, Mark Bickford and Robert L. Constable*, Symposium on Logic in Computer Science (LICS), 2017.

[5] **Formally Verified Differential Dynamic Logic**, *Brandon Bohrer, Vincent Rahli, Ivana Vukotic, Marcus Völp and Andre Platzer*, Conference on Certified Programs and Proofs (CPP), 2017.

[6] **Exercising Nuprl's Open-Endedness**, *Vincent Rahli*, International Congress on Mathematical Software (ICMS), 2016.

[7] **A Nominal Exploration of Intuitionism**, *Vincent Rahli and Mark Bickford*, Conference on Certified Programs and Proofs (CPP), 2016.

[8] **Towards a Formally Verified Proof Assistant**, *Abhishek Anand and Vincent Rahli*, International Conference on Interactive Theorem Proving (ITP), 2014.

[9] **Developing correctly replicated databases using formal tools**, *Nicolas Schiper, Vincent Rahli, Robbert Van Renesse, Mark Bickford, and Robert L. Constable*, International Conference on Dependable Systems and Networks (DSN), 2014.

[10] **Formal program optimization in Nuprl using computational equivalence and partial types**, *Vincent Rahli, Mark Bickford, and Abhishek Anand*, International Conference on Interactive Theorem Proving (ITP), 2013.

[11] **A complete realisability semantics for intersection types and arbitrary expansion variables**, *Fairouz Kamareddine, Karim Nour, Vincent Rahli, and J. B. Wells*, International Colloquium on Theoretical Aspects of Computing (ICTAC), 2008.

[12] **Uniform circuits, & boolean proof nets**, *Virgile Mogbil and Vincent Rahli*, Logical Foundations of Computer Science (LFCS), 2007.

Journals

[1] **Validating Brouwer's Continuity Principle for Numbers Using Named Exceptions**, *Vincent Rahli and Mark Bickford*, Mathematical Structures in Computer Science (MSCS), 2017.

[2] **EventML: Specification, Verification, and Implementation of Crash-Tolerant State Machine Replication Systems**, *Vincent Rahli, David Guaspari, Mark Bickford, Robert L. Constable*, Science of Computer Programming journal (SCP), 2017.

[3] **Skalpel: A Constraint-Based Type Error Slicer for Standard ML**, *Vincent Rahli, J. B. Wells, John Pirie, and Fairouz Kamareddine*, Journal of Symbolic Computation (JSC), 2016.

[4] **Reducibility proofs in the λ-calculus**, *Fairouz Kamareddine, Vincent Rahli, and J. B. Wells*, Fundamenta Informaticae, 2012.

[5] **On Realisability Semantics for Intersection Types with Expansion Variables**, *Fairouz Kamareddine, Karim Nour, Vincent Rahli, and J. B. Wells*, Fundamenta Informaticae , 2012.

Refereed workshop papers

[1] **Deconstructing MinBFT for Security and Verifiability**, *Vincent Rahli, Francisco Rocha, Marcus Völp and Paulo Esteves-Verissimo*, GRSRD, 2016.

[2] **Coq as a Metatheory for Nuprl with Bar Induction**, *Vincent Rahli and Mark Bickford*, Continuity, Computability, Constructivity Workshop (CCC), 2015.

[3] **Nuprl's Inductive Logical Forms**, *Mark Bickford, Robert L. Constable, Richard Eaton, and Vincent Rahli*, International Workshop on the use of AI in Formal Methods (AI4FM), 2015.

[4] **Formal Specification, Verification, and Implementation of Fault-Tolerant Systems using EvenML**, *Vincent Rahli, David Guaspari, Mark Bickford, and Robert L. Constable*, International Workshop on Automated Verification of Critical Systems (AVoCS), 2015.

[5] **Skalpel: A Type Error Slicer for Standard ML**, *Vincent Rahli, John Pirie, Joe Wells and Fairouz Kamareddine*, Workshop on Logical and Semantic Frameworks (LSFA), 2014.

[6] **A Generic Approach to Proofs about Substitution**, *Abhishek Anand and Vincent Rahli*, Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP), 2014.

[7] **A Type Theory with Partial Equivalence Relations as Types**, *Abhishek Anand, Mark Bickford, Robert L. Constable, and Vincent Rahli*, TYPES, 2014.

[8] **A Diversified and Correct-by-Construction Broadcast Service**, *Vincent Rahli, Nicolas Schiper, Robbert Van Renesse, Mark Bickford, and Robert L. Constable*, International Workshop on Rigorous Protocol Engineering (WRiPE), 2012.

[9] **ShadowDB: A Replicated Database on a Synthesized Consensus Core**, *Nicolas Schiper, Vincent Rahli, Robbert Van Renesse, Mark Bickford, and Robert L. Constable*, Workshop on Hot Topics in System Dependability (HotDep), 2012.

[10] **Interfacing with Proof Assistants for Domain Specific Programming Using EventML**, *Vincent Rahli*, International Workshop On User Interfaces for Theorem Provers (UITP), 2012.

[11] **The Logic of Events, a framework to reason about distributed systems**, *Mark Bickford, Vincent Rahli, and Robert L. Constable*, Languages for Distributed Algorithms workshop (LADA), 2012.

[12] **Simplified reducibility proofs of church-rosser for β- and $\beta\eta$-reduction**, *Fairouz Kamareddine and Vincent Rahli*, Workshop on Logical and Semantic Frameworks, with Applications (LSFA), 2008.

[13] **Developing realisability semantics for intersection types and expansion variables**, *Fairouz Kamareddine, Karim Nour, Vincent Rahli, and J. B. Wells*, Workshop on Intersection Types and Related Systems (ITRS), 2008.

[14] **Reducibility proofs in the λ-calculus**, *Fairouz Kamareddine, Vincent Rahli, and J. B. Wells*, Workshop on Intersection Types and Related Systems (ITRS), 2008.

Thesis and technical reports

[1] **Towards a Formally Verified Proof Assistant**, *Abhishek Anand and Vincent Rahli*, Cornell University, 2013-2014, Technical report.

[2] **Formal Specification, Verification, and Implementation of Fault-Tolerant Systems**, *Vincent Rahli, David Guaspari, Mark Bickford and Robert L. Constable*, Cornell University, 2013.

[3] **Thesis title: Investigations in intersection types: Confluence, and semantics of expansion in the λ-calculus, and a type error slicing method**, *Vincent Rahli*, Heriot-Watt University, MACS, ULTRA group, January 2011, PhD thesis.

[4] **A constraint system for a SML type error slicer**, *Vincent Rahli, J. B. Wells and Fairouz Kamareddine*, Heriot-Watt University, MACS, ULTRA group, 2010, Technical report HW-MACS-TR-0079.

[5] **Challenges of a type error slicer for the SML language**, *Vincent Rahli, J. B. Wells and Fairouz Kamareddine*, Heriot-Watt University, MACS, ULTRA group, Technical report HW-MACS-TR-0071.

Research topics

Current research topics

2018–Present **Asphalion**, Asphalion is an extension of Velisarios, also implemented in Coq, that supports the verification of hybrid fault-tolerant systems, i.e., systems that combine components that can fail arbitrarily and components that can only crash on failure. To reason about such systems, Asphalion provides a sound knowledge sequent calculus that features both non-trusted and trusted knowledge operators. Using Asphalion we have verified one of the main safety property of the MinBFT landmark protocol. (Research funded by the National Research Fund Luxembourg (FNR), through PEARL grant FNR/P14/8149128.).

2016–Present **Velisarios**, Velisarios is an extension of EventML, implemented in Coq, that supports the verification of Byzantine fault-tolerant systems, and reasoning about distributed epistemic knowledge. Using Velisarios we have verified the standard agreement safety property of the PBFT landmark protocol. In the future, we want to support proving properties such as liveness and timeliness, and we want to connect Velisarios with the Verified Software Toolchain (VST) in order to verify the correctness of programs written in C. (Research funded by the National Research Fund Luxembourg (FNR), through PEARL grant FNR/P14/8149128.).

2011–2015 **EventML**, EventML is a ML-like constructive specification language that implements a paradigm for verified programming. It gives precedence to the programming task, and also allows programmers to cooperate with a proof assistant in order to structure arguments of correctness. It is designed specifically to cooperate with the Nuprl proof assistant in order to develop correct-by-construction asynchronous protocols. Using EventML, we have specified, synthesized, and verified safety properties of the Multi-Paxos protocol. To do so, we have automated some patterns of reasoning; and to get efficient code, we have built a process optimizer in Nuprl. (Research funded by the DARPA CRASH (Clean-slate design of Resilient, Adaptive, Secure Hosts) project, award number FA8750-10-2-0238.).

2013–Present **Certified theorem proving (Nuprl in Coq)**, Allen's Partial Equivalence Relation (PER) semantics provides a semantics for Nuprl's type theory that allows one to prove that Nuprl's inference rules are valid, and therefore that Nuprl is consistent. Until recently, these proofs were done by hand. In order to formally prove that these rules are correct (and therefore that Nuprl is consistent), we have implemented this PER semantics using the Coq proof assistant. This implementation (1) provides a bridge between Nuprl and Coq, and (2) is the basis for developing a certified version of Nuprl. (Research funded by the DARPA CRASH (Clean-slate design of Resilient, Adaptive, Secure Hosts) project, award number FA8750-10-2-0238.).

2014–Present **Intuitionistic and Nominal Type Theory**, Using our formalization of Nuprl in Coq, we are turning Nuprl into an intuitionistic type theory, i.e., we have proved that some versions of Brouwer's continuity and bar induction principles are valid w.r.t. Nuprl's PER semantics. In order to prove the validity of such continuity principles we have turned Nuprl into a nominal type theory. It remains open whether stronger continuity and bar induction rules are also valid. Moreover, we have also implemented a version of Brouwer's concept of choice sequences on top of Nuprl's underlying digital library of facts and definitions. In addition, the library can now also contain sequences of choices that can be filled over time. We validated standard axioms about choice sequences by turning Nuprl's semantics into a Beth model. (Research funded by the SnT and the National Research Fund Luxembourg (FNR), through PEARL grant FNR/P14/8149128.).

2016 **Certified theorem proving (KeYmaera X in Coq)**, KeYmaera X is a theorem prover for cyber-physical systems (modeled as hybrid systems) that implements a logic called Differential Dynamic Logic (dL for short). KeYmaera X has a small core thanks a uniform substitution based proof calculus. We have implemented and verified this core in Coq. (Research funded by the SnT and the National Research Fund Luxembourg (FNR), through PEARL grant FNR/P14/8149128.).

Past research topics

2009–2011 **Skalpel**, Programming languages such as SML have sophisticated, flexible and safe type systems. Unfortunately, the type error messages for incorrect programs are confusing. Skalpel implements a promising approach to making type errors easier to understand and fix called type error slicing, in which slices (program points) containing all and only the information needed by the programmer to understand and fix a type error are identified and exhibited.

2006–2011 **Semantics of expansion**, Intersection types provide finitary type polymorphism. Expansion was introduced to recover the principal typing property in such systems. The study of realizability semantics for such systems with expansion might help cast some light on the expansion mechanism.

2006–2011 **Reducibility proofs**, Reducibility is a method based on realizability semantics where the idea is to interpret types by sets of λ-terms closed under some properties. This method seems promising in generalizing diverse properties' proofs of the (typed or untyped) λ-calculus.

2006 **Implicit complexity**, Some relations between proof nets and Boolean circuits can be expressed using the (Turing) polynomial hierarchy: there exists a proof-as-program correspondence between proof nets and (non-)deterministic Boolean circuits (using a uniform depth-preserving simulation).

Computer skills

Advanced	OCaml, SML, Coq, Agda, Nuprl, LaTeX, Git
Intermediate	C, Java, Scala, Erlang, Lisp, Haskell

Implemented Software

Velisarios A framework implemented in Coq and OCaml, to implement, execute, and reason about the safety of Byzantine fault-tolerant systems. See https://github.com/vrahli/Velisarios (~50K LOC).

Coq-dL A verified implementation of KeYmaera X's core in Coq (KeYmaera X is a proof assistant for cyber-physical systems). See https://github.com/LS-Lab/Coq-dL (~30K LOC).

NuprlInCoq A verified implementation of Nuprl in Coq. It includes a Nuprl proof translator written in OCaml. See https://github.com/vrahli/NuprlInCoq (~290K LOC).

EventML A framework to implement, execute and reason about crash fault-tolerante systems, implemented in Nuprl, SML, OCaml, Lisp, and Scala. See https://github.com/vrahli/EventML (~50K LOC).

Aneris A fault-tolerant ordered broadcast service similar to Paxos, implemented in EventML

Skalpel A static analysis tool implemented in SML that reports type error for the SML language. See https://github.com/ultra-group/skalpel (~45K LOC).

Presentations

2018 ○ **Computability Beyond Church-Turing via Choice Sequences**, LICS 2018, Oxford, UK

- **Velisarios: Byzantine Fault-Tolerant Protocols Powered by Coq**, ESOP 2018, Thessaloniki, Greece

2017
- **Bar Induction: The Good, the Bad, and the Ugly**, LICS 2017, Reykjavik, Iceland
- **Towards an Intuitionistic Type Theory**, Heriot-Watt Computer Science seminar
- **Proven-Correct Provers**, ILIAS seminar, Luxembourg University

2016
- **Exercising Nuprl's Open-Endedness**, ICMS 2016, Berlin, Germany
- **Deconstructing MinBFT for Security and Verifiability**, GRSRD 2016, Nancy, France
- **A Nominal Exploration of Intuitionism**, CPP 2016, Saint Petersburg, FL, USA

2015
- **Coq as a Metatheory for Nuprl with Bar Induction**, CCC 2015, Kochel am See, Germany
- **Formal Specification, Verification, and Implementation of Fault-Tolerant Systems using EvenML**, AVoCS 2015, Edinburgh, Scotland
- **Nuprl's Inductive Logical Forms**, AI4FM 2015, Edinburgh, Scotland

2014
- **How Trustworthy Can Systems Become?** Cornell Brown Bag seminar
- **A Type Theory with Partial Equivalence Relations as Types**, Cornell PRL seminar
- **Towards a Formally Verified Proof Assistant**, ITP 2014, VSL, Vienna, Austria
- **Developing correctly replicated databases using formal tools**, DSN 2014, Atlanta, GA, USA

2013
- **Building a verified proof assistant**, Cornell PRL seminar
- **Formal program optimization in Nuprl using computational equivalence and partial types**, ITP 2013, Rennes, France
- **Programming in Nuprl**, ENS, Paris

2012
- **A Diversified and Correct-by-Construction Broadcast Service**, WRiPE 2012, Austin, TX, USA
- **Interfacing with Proof Assistants for Domain Specific Programming Using EventML**, UITP 2012, Bremen, Germany

2011
- **A simple consensus algorithm**, Cornell PRL seminar

2010
- **Progress on Skalpel**, Heriot-Watt CS PhD seminar

2009
- **A preliminary version of Skalpel**, SPLS, Glasgow University
- **Skalpel**, Heriot-Watt CS PhD seminar

2008
- **A complete realisability semantics for intersection types and arbitrary expansion variables**, ICTAC 2008, Istanbul, Turkey
- **Lambda-calculi and Church-Rosser property**, Heriot-Watt CS PhD seminar
- **Reducibility proofs in the λ-calculus**, ITRS 2008, Torino, Italy

Teaching

Lectures

2016 Lectured for MICS 3.25 (Fault and Intrusion Tolerance) at the University of Luxembourg.

2016 Lectured for MICS 2.6 (Foundations of Computing) at the University of Luxembourg.

2015 Lectured for CS 6110 (Advanced Programming Languages) at Cornell.

2014 Lectured for CS 5860 (Introduction to Formal Methods) at Cornell.

2013 Lectured for CS 3110 (Data Structures and Functional Programming) at Cornell.

2011 Lectured for CS 5860 (Introduction to Formal Methods) at Cornell.

Certificates

2010 Obtained the status of teacher after completing LEADS3 at Heriot-Watt University.

2009 Obtained the status of tutor after completing LEADS2 at Heriot-Watt University.

2008 Obtained the status of lab demonstrator and tutorial assistant after completing LEADS1 at Heriot-Watt University.

Tutoring

2009–2010 Foundations 1: Logic and lambda-calculus, taught by Doctor Joe B. Wells at Heriot-Watt University.

2008–2010 Foundations 2: Computability (Turing machines), taught by Doctor Joe B. Wells at Heriot-Watt University.

2007–2008 Theory Bridge: Logic (proposition, predicate, Boolean interpretation), Z specification, λ-calculus, computability, taught by Professor Fairouz Kamareddine at Heriot-Watt University.

2007–2008 Logic and Proof: Logic (proposition, predicate, Boolean interpretation, quantification), taught by Professor Fairouz Kamareddine at Heriot-Watt University.

PhD & Master Supervision

2018–Present Co-supervising Cristian Mirto, a PhD student of Professor Paulo Verissimo at the SnT, working on the verification of blockchain consensus protocols.

2016–Present Co-supervising Ivana Vukotic, a PhD student of Professor Paulo Verissimo at the SnT, working on the verification of Byzantine Fault Tolerant (BFT) protocols.

2016 Co-supervised an internship project aiming at the formalization and verification of the core of the KeYmaera X proof assistant in Coq.

2016 Co-supervised a master project aiming at decomposing the MinBFT protocol, for security and verifiability.

2012–2014 Co-supervised Abhishek Anand, a PhD student of Professor Constable at Cornell.

2013 Co-supervised a master project aiming at building a translator from Nuprl to Scala.

2009–2010 Co-supervised five master projects related to our *Skalpel* project.

Community services

Conferences

2016 Program Committee member of CPP 2017.

2009 Helper at ICFP 2009.

2008 Member of the Organizing Committee of WoLLIC 2008.

Reviews

2017	CPP, EuroSys, EMSOFT, The Computer Journal
2016	ACSD, DSN, LICS, EDCC
2015	CSL
2012	ICFP
2010	RTA
2008	LSFA
2007	TYPES

Collaborators

Collaborators Abhishek Anand (Cornell), Mark Bickford (ATC-NY/Cornell), Brandon Bohrer (CMU), Liron Cohen (Cornell), Robert L. Constable (Cornell), Jérémie Decouchant (SnT, University of Luxembourg), David Guaspari (ATC-NY), Fairouz Kamareddine (Heriot-Watt), Karim Nour (Université de Savoie), John Pirie (Heriot-Watt), André Platzer (CMU), Francisco Rocha (SnT, University of Luxembourg), Robbert van Renesse (Cornell), Nicolas Schiper (Cornell), Paulo Esteves-Verissimo (SnT, University of Luxembourg), Marcus Völp (SnT, University of Luxembourg), Ivana Vukotic (SnT, University of Luxembourg), Joe B. Wells (Heriot-Watt).

PhD advisors Fairouz Kamareddine and Joe B. Wells (Heriot-Watt University).

Languages

French	**Mothertongue**
English	**Fluent**