PhD Thesis Proposal Form
China Scholarship Council (CSC)/ENS Rennes
Call for projects 2021

FIELD: Computer Science

THESIS SUBJECT TITLE: Verified programming of stream functions

1. **Single French PhD proposal:**

- Laboratory name: Inria Rennes, IRISA
- PhD director: Jean-Pierre TALPIN https://www.irisa.fr/prive/talpin/
- Position: Senior researcher (DR1)
- E-mail: talpin@irisa.fr
- Phone number: 0299847436

2. **Co-directed or a joint PhD, please specify:**

- Joint PhD (cotutelle): YES
- Partner university name: ISCAS, Beijing
- Laboratory name and web site: ISCAS' SKLCS http://lcs.ios.ac.cn/index.php/en
- PhD director in partner university: Naijun ZHAN http://lcs.ios.ac.cn/~znj/
- Position: Senior Research Professor
- E-mail: 詹乃军 znj@ios.ac.cn
- Phone: +86-10-62661615

- If previous collaborations with the Chinese co-director/university, please detail:

  Naijun Zhan and I collaborate since 2015, first in the context of mutual scientific visits in Beijing and Rennes and, since 2017, in the context of the Inria associate team CONVEX (http://convex.irisa.fr). Our main topic of collaboration is the compositional verification of cyber-physical systems. We have jointly published two conference and journal papers and are currently submitting three other ones, especially in the context of a joint post-doctorate researcher: Xiong XU, currently funed by Inria and ISCAS on our joint project.

- Interest of the Joint PhD for the French co-director, for his/her laboratory, for ENS Rennes:

  ENS-Rennes and Inria Rennes are both academic partners of IRISA, and I actively collaborate with ENS-Rennes, currently co-advising two PhD students with senior faculties of ENS-Rennes, and as participant to the elaboration of collaboration frameworks with ECNU, Shanghai, and the Shenyuan College of BUAA, Beijing.

- PROPOSAL

  Proof and type theories and formal verification by satisfaction modulo theory (SMT) have led to revisit the paradigm "proof = program" of the lambda-calculus by introducing the notion of type refinement and decidable dependent type theory. This scientific development enables verified programming: halfway between deductive programming and synthesizing programs from theorem provers. A refinement <v: t | p> defines the domain of v by its data type t but refines its domain of definition by the logical property p. For instance, the dependent type <n: int | n mod 2 = 0> denotes even integers, <n: int | n * 2 = m> defines n as the double of m, etc.

  The paradigm of verified programming is implemented by means of algorithmic languages like Liquid Haskell and F* and is, to a certain extent, also present with imperative languages like Frama-C. Its use makes it possible to certify the correctness of a program with respect to requirements expressed by means of dependent types, for example security requirements (cryptographic protocols), by using verification (Z3) or proof (Coq) tools.

  The goal of our project is to revisit the principles of stream functions and data-flow programming by embedding these models of computation as domain-specific formalisms using verified programming languages Liquid Haskell (UCSD, IMDEA) and F * (Inria-Microsoft) and to support both discrete (for example FRP) and continuous (Yampa) systems.

  Ultimately, and within this framework, we would like to support the generation of reactive (non-preemptive, static scheduling) or hybrid (preemptive execution, dynamic scheduling) programs from this model (using existing compilers such as F*'s Low* or Steel), thus obtaining a verified modeling paradigm for the intended application area of hybrid system design.

  The start of the project will be to develop the basic building blocks of the model, while keeping an aim on its possible continuation for model verification (Simulink, HCSP) and/or program synthesis (Steel, Rust), by defining a reactive programming paradigm within F* and its necessary extensions to model cyber-physical systems.

  The development of our project will be driven by case studying practical applications these concepts within Inria's associate project Convex with ISCAS, such as the modeling and type-based verification of Simulink/Stateflow/AADL data-flow diagrams, the use of hybrid modeling and verification tools such as HCSP and HHL in Isabelle, the synthesis of Rust programs.

  REFERENCES

  F*: A Higher-Order Effectful Language for Program Verification. https://www.fstar-lang.org
  FRP: Practical Functional Reactive Programming. https://reflex-frp.org
  Yampa: https://github.com/ivanperez-keera/Yampa
  LTL types FRP: https://dl.acm.org/citation.cfm?id=2103783
  Steelcore: https://www.fstar-lang.org/papers/steelcore
  HHL : An improved HHL prover: An interactive theorem prover for hybrid systems, S. Wang, N. Zhan and L. Zou. ICFEM'15, Lecture Notes in Computer Science n°9407.
  HCSP: Modelling and Verifying Communication Failure of Hybrid Systems in HCSP. S. Wang, F. Nielson, H. Nielson and N. Zhan. Computer Journal, 2017.

- PUBLICATIONS

  "Parallel Composition and Modular Verification of Computer-Controlled Systems in Differential Dynamic Logic". S. Lunel, S. Mitsch, B. Boyer, and J. Talpin. 23rd International Symposium on Formal Methods. Springer, 2019.
  "Unified Graphical Co-Modelling of Cyber-Physical Systems using AADL and Simulink/Stateflow". H. Zhan, Q. Lin, S. Wang, J. Talpin, X. Xu, and N. Zhan. 7th International Symposium on Unifying Theories of Programming. Springer, 2019.
  "Towards verified programming of embedded devices". J.-P. Talpin, J.-J. Marty, S. Narayan, D. Stefan, R. Gupta. Invited paper at the 23rd Design, Automation and Test in Europe. IEEE, 2019.
  "Indecision and delays are the parents of failure – Taming them algorithmically by synthesizing delay-resilient control. M. Chen, M. Fraenzle, Y. Li, P. Mosaad, N. Zhan. Acta Informatica, 2020.