Sophie Fortz, Postdoctoral Researcher

sophie.fortz@kcl.ac.uk
@FortzSophie
https://sfortz.github.io/
https://orcid.org/0000-0001-9687-8587

The long term goal of my research is to enhance software reliability through the automation of verification and validation tasks. Leveraging my expertise in behavioural inference, I aim to design model-based approaches tailored for complex systems such as variability-intensive, quantum, and AI-based software.
During my Ph.D. at the University of Namur, I inferred behavioural models for variability-intensive systems, *i.e.,* Featured Transition Systems. This research builds bridges between software product lines, automata learning, and deep learning research communities. This Ph.D. was partly funded by the FNRS EoS (Excellence of Science) VeriLearn project, before I obtained a FRIA (FNRS) Grant. In November 2023, I joined King's College London as a postdoctoral researcher on the VSL-Q project. Our mission within this project is to provide verified simulations for large-scale quantum systems, tackling practical challenges such as platform specificities, noise, and scalability. This collaborative effort brings together experts from diverse fields including physics, programming languages, and software engineering.

Research Interests

Software Variability	Software product line engineering, configurable processes, variability-intensive system behaviour, featured transition systems, variability mining.
Automata Learning	Active automata learning, behavioural model learning, software reverse engineering.
Quantum Computing	Quantum computing, verified software, high-level modelling.
Artificial Intelligence	Deep Learning, symbolic AI, AI for SE, SE for AI.

Employment History

Nov. 2023 – *Ongoing*	**Postdoctoral Researcher.** Software Systems group, Department of Informatics, Faculty of Natural, Mathematical & Engineering Sciences, King's College London, United Kingdom.
Oct. 2023 – Nov. 2023	**Postdoctoral Researcher.** PRECISE, NaDi, Faculty of Computer Science, University of Namur, Belgium.
Oct. 2020 – Sept. 2023	**PhD Student under FRIA Grant (FRS-FNRS).** PRECISE, NaDi, Faculty of Computer Science, University of Namur, Belgium.
Sept. 2019 – Sept. 2020	**PhD Student.** PRECISE, NaDi, Faculty of Computer Science, University of Namur, Belgium.

Education

2019 – 2023 **Ph.D. in Software Engineering.**
Thesis title: *Learning Featured Transition Systems.* Supervisors: Dr. Gilles Perrouin & Prof. Patrick Heymans. University of Namur, Namur, Belgium.

2017 – 2019 **M.Sc. Computer Science. (Magna Cum Laude).**
Software Engineering specialty. Thesis title: *SAT-Based Concolic Testing in Prolog.* Supervisors: Prof. Wim Vanhoof. University of Namur, Namur, Belgium.

2014 – 2017 **B.Sc. Computer Science.) (Cum Laude).**
Mathematics and English options, University of Namur, Namur, Belgium.

Teaching

2019-2023 **Software Testing:** project supervision, MSc Level, ± 15 students, University of Namur.

2019–2022 **Mathematics Fundamentals for Computer Science:** exercise sessions on recurrent equations and cryptography basics, BSc Level, $2 \times \pm 40$ students, University of Namur.

2020-2021 **Introduction to the Scientific Approach:** one group project supervision, 2021, BSc Level, 2 students, University of Namur.

Recent Projects

QAssure **Assurance of Quantum Technology Systems.** Innovate UK-funded project under the "Scalable Quantum Network Technologies: Collaborative R&D" programme. I am part of this consortium, led by BT Group, to develop methodologies for assuring quantum systems such as Quantum Key Distribution (QKD). 2024–2025. Partners: BT Group (coordinator), National Physical Laboratory (NPL), University of Bristol, Cystel, HSBC, KETS Quantum Security, King's College London, Loughborough University, NodeQ, Toshiba Europe Ltd, and Quentangle. Keywords: Quantum Key Distribution, Quantum Assurance, Cybersecurity.

VSL-Q **Verified Simulation for Large Quantum Systems.** This project brings together researchers from software engineering, programming languages, and quantum physics to address the increasing complexity of quantum computing systems and the stringent reliability demands of future applications. The goal is to create a verified end-to-end framework that connects high-level programming languages to low-level implementations on various physical quantum platforms. VSL-Q proposes a layered architecture with verified mappings between layers, ensuring reliable compilation from high-level programs to low-level, reliability-aware implementations. A key case study involves simulating fermionic systems to evaluate the approach's scalability. 2023–2025. Partners: King's College London (coordinator), University of Oxford. Keywords: Quantum Computing, Verified Software, High-level Modelling.

Recent Projects (continued)

RoaRQ

- **Robust and Reliable Quantum Computing.** Funded by the Engineering and Physical Sciences Research Council, this project brings together a cross-disciplinary community of researchers in quantum computing and computer science to tackle the challenge of delivering robust, reliable, and trustworthy quantum computing. The programme aims to establish a framework for quantum computation verification across key layers of quantum computing stacks. One key investigation, *ModeMCQ*, focuses on model-based monitoring and calibration of quantum computations, targeting the development of platform-independent programming paradigms for NISQ algorithms with automatic error mitigation strategies. 2023–2026. ModeMCQ Partners: King's College London (KCL), University of Oxford. Keywords: Quantum Computing, NISQ, Error Mitigation, Quantum Verification.

LIFTS

- **Learning Featured Transition Systems.** This PhD project, funded by a competitive FRIA Grant from the FRS-FNRS, aimed to advance the learning of Featured Transition Systems (FTS), a formalism used for modelling the behaviour of variability-intensive systems. The research focused on combining automata learning techniques with deep learning to efficiently explore and understand complex, variable-rich systems. 2019–2023. Keywords: Featured Transition Systems, Automata Learning, Deep Learning.

VeriLearn

- **Verifying Systems that Learn.** The goal of this project is to integrate the disciplines of software verification and machine learning to provide strong guarantees for adaptive software systems. VeriLearn addresses key questions around how to verify systems that modify themselves through learning, ensuring they meet ethical, privacy, and societal concerns. The project focuses on developing techniques to combine logic, probability, and machine learning, with an emphasis on interpretable models like probabilistic logic programming. This research aims to establish a framework for mathematically verifiable AI systems. Excellence of Science (EoS) project, 2018–2023. Partners: KU Leuven (coordinator), UNamur, ULB. Keywords: Machine Learning, Software Verification, AI Safety.

Distinctions & Awards

2024

- Distinguished Reviewer Award, 28th International Systems and Software Product Line Conference

Community Service

Organizing committee

- **Q-SE**: International Workshop on Quantum Software Engineering (Web & Publicity chair), co-located with ICSE, 2025.
- **QSANER**: International Workshop on Quantum Software Analysis, Evolution and Reengineering (co-Chair), co-located with SANER, 2025.
- **ICTSS**: International Conference on Testing Software and Systems (Challenge Track co-Chair), 2024.
- **SPLC**: ACM Software Product Line Conference (Publicity chair), 2024.

Community Service (continued)

- **LearnAut**: Learning and Automata (Co-chair), co-located with ICALP / LiCS / FSCD, 2024.

Program Committees

- **GI ICSE**: International Workshop on Genetic Improvement, co-located with ICSE, 2025.
- **ICST**: IEEE International Conference on Software Testing, Verification and Validation (Poster track), 2025.
- **VaMoS**: International Working Conference on Variability Modelling of Software-Intensive Systems, 2025.
- **BENEVOL**: Belgium-Netherlands Software Evolution Workshop, 2024.
- **ICTAC**: International Colloquium on Theoretical Aspects of Computing (Research track), 2024.
- **SPLC**: ACM System and Software Product Line Conference, 2022 (Research track, as a sub-reviewer), 2023 (Demonstration & Tools) and 2024 (Research track).
- **SSBSE**: Symposium on Search-Based Software Engineering (Challenge Track), 2024.
- **ECOOP**: European Conference on Object-Oriented Programming (Artefacts, Extended Review Committee), 2024.
- **ESEC-FSE**: ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (Artefacts), 2023.
- **ICSR**: International Conference on Software and Systems Reuse (Research track, as a sub-reviewer), 2022.

Journal Reviewing

- **TSE**: Transactions on Software Engineering. Publisher: IEEE.
- **JSS**: Journal of Systems and Software. Publisher: Elsevier.
- **SoSyM**: International Journal on Software and Systems Modeling. Publisher: Springer.
- **SQJ**: Software Quality Journal. Publisher: Springer.
- **EMSE**: Empirical Software Engineering (as a sub-reviewer). Publisher: Springer.

Community Service (continued)

Other Community Services

- **Master Thesis Juries**: Member of several juries for master thesis;
- **Computer Science Faculty Council**: Student representative, then scientific representative.
- **Students Fairs**: Representing and promoting the Computer Science Faculty at several Students Fairs;

Miscellaneous Experience

2019–2023 — **ALMIN board member and president**, University of Namur, Belgium. ALMIN is the alumni association of the Computer Science Faculty at the University of Namur, dedicated to fostering connections between graduates and supporting the student community. As a board member, I contributed to organising events and networking opportunities for alumni and students. I was a board member of the association since my graduation in 2019, and took its presidency from 2021 to 2023.

2019 — **Research internship**, Department of Computer Systems and Computation, Polytechnic University of Valencia, Spain. During my master thesis, I have done three months of research at the polytechnic university of Valencia (Spain), under the supervision of Prof. German Vidal. My work on concolic testing for logic programming was nominated for the Jean Fichefet award (best master thesis award).

2017 – 2019 — **CSLabs secretary and board member**, Computer Science Labs (CSLabs), Namur, Belgium. CSLabs is a non-profit organisation founded by students from Unamur's Faculty of Computer Science, in order to promote computer science externally and provide trainings in different fields by and for students.

Languages

French	Native (mother tongue).
English	B2-level (score of 72 on the Pearson PTE Academic test, 2023).
Dutch	Basic knowledge.
French Belgian Sign Language	Basic knowledge.

Publications

1. D'Aloisio, G., **Fortz, S.**, Hanna, C., Fortunato, D., Bensoussan, A., Usandizaga, E. M., & Sarro, F. (2024). Exploring LLM-Driven Explanations for Quantum Algorithms. *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM.*
2. **Fortz, S.**, Temple, P., Devroey, X., Heymans, P., & Perrouin, G. (2024). VaryMinions: Leveraging RNNs to Identify Variants in Variability-intensive Systems' Logs. *Empirical Software Engineering*.
3. **Fortz, S.**, Temple, P., Devroey, X., & Perrouin, G. (2024). Towards feature-based ml-enabled behaviour location. *Proceedings of the 18th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS)*, 152–154.

4. **Fortz, S.** (2023). Variability-aware behavioural learning. *Proceedings of the 27th ACM International Systems and Software Product Line Conference (SPLC) - Volume B*, 11–15.

5. dos Santos, E. L., **Fortz, S.**, Schobbens, P., & Perrouin, G. (2022). Identifying architectural smells in self-adaptive systems at runtime. *13ème édition de la Conférence francophone sur les Architectures Logicielles (CAL)*.

6. dos Santos, E. L., **Fortz, S.**, Perrouin, G., & Schobbens, P. (2021). A vision to identify architectural smells in self-adaptive systems using behavioral maps. *4th Context-aware, Autonomous and Smart Architectures International Workshop (CASA@ECSA), 2978*.

7. dos Santos, E. L., **Fortz, S.**, Schobbens, P., & Perrouin, G. (2021). Behavioral maps: Identifying architectural smells in self-adaptive systems at runtime. *Software Architecture, 13365*, 159–180.

8. **Fortz, S.** (2021). LIFTS: learning featured transition systems. *Proceedings of the 25th ACM International Systems and Software Product Line Conference (SPLC) - Volume B*, 1–6.

9. **Fortz, S.**, Temple, P., Devroey, X., Heymans, P., & Perrouin, G. (2021). VaryMinions: Leveraging RNNs to identify variants in event logs. *Proceedings of the 5th International Workshop on Machine Learning Techniques for Software Quality Evolution (MaLTeSQuE@ESEC/FSE)*, 13–18.

10. **Fortz, S.**, Mesnard, F., Payet, É., Perrouin, G., Vanhoof, W., & Vidal, G. (2020). An SMT-based concolic testing tool for logic programs. *15th International Symposium on Functional and Logic Programming (FLOPS), 12073*, 215–219.

Bibliometrics

Total Number of Publications:	**10**
Total Number of Citations:	**31**
H-index (Google Scholar):	**4**

See `https://scholar.google.co.uk/citations?user=cfV6X6kAAAAJ` for the full list of publications.

Ph.D. Thesis

Title — Learning Featured Transition Systems Learning Featured Transition Systems

Supervisors — Dr. Gilles Perrouin and Prof. Dr. Patrick Heymans

Year — 2023

Ph.D. Thesis (continued)

Abstract

Variability-intensive Systems (VISs) are software-based systems whose characteristics and behaviour can be modified by the activation or deactivation of some options. Addressing variability proactively during software engineering (SE) activities means shifting from reasoning on individual systems to *reasoning on families of systems*. Adopting appropriate variability management techniques can yield important *economies of scale* and quality improvements. Conversely, variability can also be a curse, especially for Quality Assurance (QA), *i.e., verification and testing* of such systems, due to the combinatorial explosion of the number of software variants. Indeed, by combining only 33 Boolean options, we can define more variants of a system than the number of people on Earth. Verifying or testing each variant individually is thus impossible in most practical cases.

About a decade ago, *Featured Transition Systems (FTSs)* were introduced as a formalism to represent, and reason on, *the behaviour of VISs*. Instead of representing each variant by a (classical) transition system, an FTS bears annotations that relate transitions to options through *feature expressions*. FTSs thus make it possible to reason at the family level by modelling all the variants of a system in a single behavioural model. FTSs have been shown to significantly improve the possibilities and execution time of automated QA activities such as model-checking and model-based testing. They have also shown their usefulness to guide design exploration activities. Yet, as most model-based approaches, FTS modelling requires both *strong human expertise and significant effort* that would be unaffordable in many cases, in particular for large legacy systems with outdated specifications and/or systems that evolve continuously.

Therefore, this thesis aims *to automatically learn FTSs from existing artefacts*, to ease the burden of modelling FTS and support continuous QA activities. To answer this research challenge, we propose a two-phase approach. First, we rely on deep learning techniques to locate variability from execution traces. For this purpose, we implemented a tool called *VaryMinions*. Then, we use these annotated traces to learn an FTS. In this second part, we adapt the seminal L^* algorithm to *learn behavioural variability*. Both frameworks are open-source and we evaluated them separately on several datasets of different sizes and origins (*e.g.,* software product lines and configurable business processes).

Keywords

Variability-intensive Systems, Software Product Line, Featured Transition Systems, Reverse Engineering, Active Automata Learning, Variability Mining

Master Thesis

Title

SAT-Based Concolic Testing in Prolog

Supervisor

Prof. Dr. Wim Vanhoof

Year

2019

Master Thesis (continued)

Abstract

Concolic testing has been studied for years in the field of imperative programming. However, we can only find a very few cases where this technique is applied to other paradigms, like logic programming. This master thesis aims at presenting a full method to apply both concrete and symbolic execution in parallel on Prolog programs. Our approach is based on a new definition of path coverage, specific to logic programming and called "choice coverage". This criteria was defined for the first time by Mesnard et al. (2015). We also introduce a prototype implementation of our algorithm.

Keywords

Concolic execution, symbolic execution, software testing, choice coverage, Prolog, logic programming.