Advanced debuggers for concurrency
Extendable virtual machines for advanced debugging support
Back-in-time and scriptable debuggers
of the following topics:

The goal of this Postdoc is to build new advanced debuggers to explore and experiment one or more:

Objectives
Mission confiée

Back-in-time and scriptable debuggers
• How to build back-in-time debuggers?
• What abstractions do we need to take advantage of back-in-time debuggers? (scripting, DSLs,)
• How can users extend, customise and enhance their debugger? (following PhD of A. Chis on Moldable Tools and the work of T. Dupriest on scriptable debuggers)

Extendable virtual machines for advanced debugging support
• What primitives do we need from the virtual machine to support advanced debugging tools?
• (back-in-time debuggers, object-centric debuggers, etc.)
• How can users create new debugging primitives and enable them on-demand when a particular bug happens and requires investigation without restarting the program?

Advanced debuggers for concurrency
• How can we track specific objects in a concurrent execution and how this capability helps in debugging concurrent programs?
• (following the PhD of S. Costiou on unanticipated object-centric debugging)
• What are and how to express object-centric debugging queries on the flow of objects, shared by concurrent processes?
• Can we leverage Dynamic Software Update techniques to safely instrument running
Support for debugging tools

- Can we use type inferencers to ease the design of debugging tools and limit their impact on performance and memory on dynamic languages runtimes? (following work on RoelTyper and PhD of L. Spoon on type inference)

References:

2. Pablo Tesone, Guillermo Polito, Noury Bouraqadi, Stéphane Ducasse, Luc Fabresse. Dynamic Software Update from Development to Production. The Journal of Object Technology, Chair of Software Engineering, 2018
9. S. A. Spoon and O. Shivers, Demand-Driven Type Inference with Subgoal pruning: Trading Precision for Scalability, Proceedings of ECOOP'04, 2004

Principales activités

The plan is to:
- Join the team work around debugging, that includes 3 researchers and 1 phd student
- Survey the key developer activities
- Define new model and enhance current
- Realize prototypes, possibly using Pharo
- Run validation with real end-users
- Publish results in top venues

Compétences

Technical skills and level required
- OOP, TDD
- Reflective programming
- Program transformation

Language
- English

Relational skills
- Good team work skills

Other valued appreciated
- Knowledge of the Pharo language is a plus
- A strong interest about debugging and/or dynamic languages is also appreciated

Avantages
- Subsidized meals
- Partial reimbursement of public transport costs
- Leave: 7 weeks of annual leave + 10 extra days off due to RTT (statutory reduction in working hours) + possibility of exceptional leave (sick children, moving home, etc.)
- Possibility of teleworking (after 6 months of employment) and flexible organization of working hours
- Professional equipment available (videoconferencing, loan of computer equipment, etc.)
- Social, cultural and sports events and activities
- Access to vocational training

Rémunération

Gross monthly salary (before taxes): 2 653 €