Automatic Synthesis of Self-Healing Android Applications

Keywords: Programming Languages, Software Engineering, Static and Dynamic Analysis.

Where. The internship will be at the Computer Science Laboratory of the École Polytechnique (LIX), in the Cosynus research team under the supervision of Sergio Mover (from LIX) and Simon Bliudze (from INRIA Lille — Nord Europe).

Project Description. Developing reliable Android applications is difficult and error-prone. Developers struggle to write the asynchronous code executed when receiving an event (e.g., touching the screen of the device). A challenge for the developers is to think about what happens when the application runs with different orders of events. In an Event Driven Framework like Android such order of events depends on the implementation of the framework. Moreover, such order is not static but changes dynamically depending on the execution of the developer's code. For example, the developer could explicitly disable the execution of the app code that respond to a click event of a button. An event-driven framework contains much more involved and complex examples showing how the *control-flow* of the app changes dynamically. The developer can also violate the implicit application-framework protocol when invoking a new framework method. As an example, in Android a developer is not supposed to dismiss a dialog object when it's enclosing activity object is already "paused".

The goal of the internship is to design and implement "self-healing" techniques for Event-Driven frameworks. Given an Android app, we want to automatically synthesize a new version of the app that avoids protocol violations — for example removing particular interleavings of events. While similar approaches exist for data-race errors [7, 1], no techniques exist to tackle protocol violations for event-driven frameworks.

Objectives The main objective of the internship is to define the self-healing problem and the synthesis algorithm, implement the algorithm (including the instrumentation of existing Android applications), and evaluate their efficacy and performance on real Android apps. The approach will rely on previous works on Android framework models [6, 5], static analysis techniques (e.g., [2, 4]), and component-based design for Java [3].

Application Applicants should send an email to Sergio Mover (sergio.mover@lix.polytechnique.fr) and Simon Bliudze (simon.bliudze@inria.fr) with a curriculum vitae and a motivation letter.

Contact. For further information and details about the internship please contact Sergio Mover via email (sergio.mover@lix.polytechnique.fr).

References

[1] Christoffer Quist Adamsen, Anders Møller, Rezwana Karim, Manu Sridharan, Frank Tip, and Koushik Sen. Repairing event race errors by controlling nondeterminism. In *Proceedings of the 39th International Conference on Software Engineering*, pages 289–299. IEEE Press, 2017.

[2] Sam Blackshear, Bor-Yuh Evan Chang, and Manu Sridharan. Selective control-flow abstraction via jumping. *ACM SIGPLAN Notices*, 50(10):163–182, 2015.

[3] Simon Bliudze, Anastasia Mavridou, Radoslaw Szymanek, and Alina Zolotukhina. Exogenous coordination of concurrent software components with javabip. *Softw., Pract. Exper.*, 47(11):1801–1836, 2017.

[4] Roberto Cavada, Alessandro Cimatti, Michele Dorigatti, Alberto Griggio, Alessandro Mariotti, Andrea Micheli, Sergio Mover, Marco Roveri, and Stefano Tonetta. The nuxmv symbolic model checker. In *Computer Aided Verification*, pages 334–342, 2014.

[5] Shawn Meier, Aleksandar Chakarov, Maxwell Russek, Sergio Mover, and Bor-Yuh Evan Chang. Abstracting event-driven systems with lifestate rules. *CoRR*, abs/1701.00161, 2017.

[6] Arjun Radhakrishna, Nicholas V. Lewchenko, Shawn Meier, Sergio Mover, Krishna Chaitanya Sripada, Damien Zufferey, Bor-Yuh Evan Chang, and Pavol Cerný. Droidstar: callback typestates for android classes. In *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018*, pages 1160–1170, 2018.

[7] Guy Martin Tchamgoue, Kyong Hoon Kim, and Yong-Kee Jun. Eventhealer: Bypassing data races in event-driven programs. *Journal of Systems and Software*, 118:208–220, 2016.