Project title: Analysis of non-functional properties for concurrent systems

Olaf Owe, Martin Steffen, Christian Johansen, Volker Stolz

March 11, 2016

1 Main objective and summary of the project:

The goal of the project is to develop and apply *static analysis and dynamic enforcement techniques and tools* to enhance the robustness of modern concurrent software, with a special focus on non-functional properties, in particular robustness- and security-related properties. The position will

1. work out new formal analysis techniques combining run-time and/or compile time approaches,
2. be accompanied by verification and *tool-supported* analysis techniques,
3. be validated prototypically on existing applications.

2 Project background and scientific basis.

Concurrency and multi-threading have become mainstream even standard users are indirectly exposed, be it in the form of multicore-computers and even multicore smart phones and tables, or in the form of cloud applications. Also modern programming languages like Google's Go, Rust, Clojure, Erlang, Scala, and Creol/ABS. Apple's Swift language, to name a few, are without exception designed with concurrency at the core, as opposed to as an afterthought.

3 Research questions and scientific challenges

On the one hand, assuring robustness for concurrent and distributed system is more pressing: getting such programs corrects is notoriously hard. This

may be attributed to the lack of appropriate abstraction, to the added complexity dimension of interactive and parallel executions —whole new classes of errors are possible only under concurrent execution— or simply lack of training of programmers. On the other hand: the singly most widely technique in practice of assuring quality in software, *testing* in all if its forms, while remaining indispensable, faces serious limitations. The most serious being that errors become irreproducible (or at least hard to reproduce), due to non-deterministic behavior.

4 Scientific method

The project will use and develop *formal methods*, i.e., methods on rigorous semantical foundation which allow to reason mathematically about the consequences of system designs. Since static and dynamic analysis techniques have complementary strengths (and weaknesses) concerning precision, computational overhead, etc. the project will aim at combining them. Techniques we plan to use include constraint solving, advanced flow analysis, and type-based techniques.

5 Project timeline

First semester is dedicated to literature study and the collection of background information. This will be supported by a weekly seminar with the supervisors and other interested PhD students and researchers. The second semester is dedicated to writing a conference paper jointly with the supervisors. Here the PhD student may be in charge of a case study. In the second year the PhD student will work on further papers and start implementation work. The last two years include on further publications, take a leading role, and on implementation work.

Teaching duties will be done mainly during the 3 middle years. Course work will be done during the 3 first years.

6 Institutional embedding and collaboration

The PhD position is planned to be connected to the research group for Precise Modelling and Analysis (PMA), with it's traditional focus of program verification and analysis, system modelling, and innovative language design. As far as the analysis of security aspects is concerned, the project will profit

from collaboration with the Strategic Research Initiative for Concurrent Security and Robustness for Networked Systems ConSeRNS. Further projects with overlapping research focus are

- GoReTech (Go Runtime Enforcement techniques, Collaboration with E. Bodden (Paderborn University)
- Cost action ARVI (Run-time verification beyond Monitoring)
- IOTSEC (Security in IoT for Smart Grids)