Katharina Mehner

Trace-based Debugging and Visualisation of Concurrent Java Programs with UML

Abstract

This thesis describes an approach for automated detection of concurrent liveness failures in the execution of Java programs.

Concurrent programs are highly prone to failure because of the inherent nondeterminism. Developers of concurrent Java programs are not well supported in detecting concurrency failures, i.e. failures that are due to interactions between multiple threads. These failures are neither well documented nor do tools like debuggers allow developers to identify them at runtime.

This thesis analyses and classifies liveness failures, a special kind of concurrency failures, and the associated potentials in Java. A UML statechart is developed that models the interaction of Java threads. Liveness failures and potentials are specified formally in terms of the states controlling the interaction of threads and in terms of the events exchanged by interacting threads.

Detection algorithms are developed to identify the specified failures in a program execution. A UML profile extending UML interaction diagrams is developed to visualise the execution of concurrent Java programs and detected liveness failures and potentials.

In order to deploy the algorithms and the UML profile, tool support concepts are provided. This involves the specification of a trace format and a tracing method to collect execution data from a running Java program, and the specification of methods to analyse the trace and to visualise the trace and the analysis results.

The concepts are implemented in the JAVIS prototype, which consists of a Java tracer with an analysis facility for monitoring liveness in concurrent Java programs, and a plug-in extension to the UML CASE tool Together for importing and displaying concurrent Java traces including failures and potentials.