

# Incremental, two-level deadlock analysis for incomplete Java Card 3.0 programs<sup>\*</sup>

Rebekka Neumann

University of Paderborn,  
Fuerstenallee 11, 33102 Paderborn, Germany  
`rneumann@s-lab.upb.de`

**Abstract.** We present an approach that combines static program analysis and model checking to verify software that resides on smart cards. The Java Card 3.0 standard (JC 3.0) introduces concurrency as a new language feature in Java for smart cards. Hence, we have chosen deadlock detection as our first goal for verification.

A web server with its libraries burned on JC 3.0 smart cards is a realistic object for our analysis. From the viewpoint of analysis it is to be considered as an incomplete program that can be completed by JC 3.0 web applications loaded on the smart card. The costs of our deadlock analysis will be reduced by analysing the web server separately and by reusing the results when the completed program is analysed.

Since results of static program analysis are conservative, some of them indicate only potential deadlocks. We will further explore these cases by model checking to get definite answers. The presented analysis approach cannot substitute testing completely in the quality assurance process, but it can simplify and guide the testing activities for deadlock detection in JC 3.0 programs.

As a first result, we show that 50% of the static analysis results for the JC 3.0 web server will be reusable when analysing the completed program.

**Key words:** Deadlock detection, static program analysis, software model checking, testing process, incomplete programs

---

<sup>\*</sup> This abstract is part of a short paper of the same title published at TAP 2009.