Software Engineering Seminar (WS 2015)

Run-time Verification of Non-functional Properties

Description

Runtime verification is concerned with monitoring and analysis of software or hardware system executions. The field is often referred to under different names, such as runtime verification, runtime monitoring, runtime checking, runtime reflection, runtime analysis, dynamic analysis, runtime symbolic analysis, trace analysis, log file analysis, etc. A running system can be abstractly regarded as a generator of execution traces, i.e., sequences of relevant states or events. Traces can be processed in various ways, e.g., checked against formal specifications, analyzed with special algorithms, visualized. The student is supposed to focus on software run-time verification techniques and investigate the state of the art. Runtime verification domains also should be presented. The student should understand the need of run-time verification and be able to interpret in which domains it is applicable and why is important for self-adaptiveness.

References

[1] Antonio Filieri, Carlo Ghezzi, and Giordano Tamburrelli. Run-time efficient probabilistic model checking. In *Proceedings of the 33rd international conference on software engineering*, pages 341–350. ACM, 2011.

[2] Martin Leucker and Christian Schallhart. A brief account of runtime verification. *The Journal of Logic and Algebraic Programming*, 78(5):293–303, 2009.

[3] Ji Zhang and Betty HC Cheng. Model-based development of dynamically adaptive software. In *Proceedings of the 28th international conference on Software engineering*, pages 371–380. ACM, 2006.

Contacts

Lars Grunske (`grunske@informatik.hu-berlin.de`)
Software Engineering Group
Institut für Informatik
Humboldt-Universität zu Berlin