



Subject Area

- Dynamic Architectures,
- Runtime Verification.

Background

A *dynamic architecture (DA)* is an architecture in which components as well as connections between their ports may change over time. A simple example of a DA is the object model of an object-oriented system in execution.

Runtime verification (RV) is a dynamic analysis method aiming at checking whether a run of the system under scrutiny satisfies a given correctness property. The inputs to a RV system are: (1) a system to be checked, and (2) a set of properties to be checked against the system execution. The properties can be expressed in a formal specification language (e.g., automata-based or logic-based formalism), or even as a program in a general-purpose programming language.

Objectives

The main objective of this project is to apply runtime verification to check whether a given software system satisfies certain architectural constraints.

Therefore, the following tasks have to be performed:

- 1) Analyse the architecture of a given software system.
- 2) Propose a set of architectural constraints the system should adhere to.
- 3) Implement the corresponding checks.
- 4) Verify whether the systems architecture adheres to the constraints.

Additional Information

- If desired, the student has the possibility to participate in a follow-up publication at a scientific venue.

Prerequisites

- Knowledge in modelling of distributed systems.
- Basic experience of a programming language such as Java or C#.
- Interest in Software Architectures.

Further References

- Runtime Verification: <http://www.havelund.com/Publications/rv-tutorial-ios-2012.pdf>

Supervisor

Prof. Dr. Dr. h.c. Manfred Broy

Advisors

Diego Marmsoler, TU München (diego.marmsoler@tum.de)

Ana Petrovska, TU München (ana.petrovska@tum.de)