Continuous Integration of Performance Models and Frameworks for Data Analysis

Background. Performance models of software systems can be used to anticipate how a system will react to changes in its structure, usage or environment. It is, however, expensive to build accurate prediction models manually. At the Architecture-Driven Requirements Engineering Group (ARE), we are working on an approach for the automated extraction of parameterized models from source code that are updated when the underlying source code or the system's environment changes. This process is then tightly integrated with deployment and operation (DevOps) to parametrize the models based on dynamic analysis and ensure that accurate models are available at all times during the software development process.

Problem. Currently, our approach is focused on extracting performance models from source code and targeted source code instrumentation and monitoring of the system for updating and calibrating those models. This extraction does not differentiate between communication inside the system and the usage of "external" components such as libraries, middleware, frameworks or external services. This can be disadvantageous, e.g., when the monitoring of those services cannot be done by source code instrumentation, because the code is not available. Plus, external components can have specific monitoring interfaces that can be used for monitoring and calibration instead of additional monitoring overhead.

Goals. For the aforementioned reasons, we aim to enrich the model extraction to allow the detection of calls to external components which are then transformed into appropriate representations in the performance model. They need to be handled differently when deciding which parts of the system to instrument or monitor, and how to use the resulting measurements for calibration. Your task is to devise a flexible method for this incorporation and evaluate it using an example system.

Prerequisites.

- Solid Java coding skills and knowledge object-orientation
- Interest in software engineering research and methods and model-driven software engineering
- Additional knowledge in software architecture and software design are helpful, but not mandatory