

Master / Bachelor

Kontinuierliche Extraktion und Analyse von Softwarearchitekturmodellen von Open Source Software

Continuous Extraction and Analysis of Software Architecture Models in Open Source Software

Motivation — Deutsch

Softwarearchitektur beschreibt die wichtigsten Designentscheidungen für ein Softwaresystem auf einer hohen Abstraktionsebene. Die Beschreibung der Softwarearchitektur ist ein wesentlicher Treiber für die Produktqualität. Viele Softwaresysteme, einschließlich Open-Source-Systeme, haben keine aktuellen Architekturbeschreibungen.

In dieser Arbeit werden Sie den Quellcode und die Dokumentation von Open-Source-Softwaresystemen nach der zugrunde liegende Software-Architektur untersuchen. Sie werden Tools zur automatischen Generierung von Software-Architekturmodellen aus dem Quellcode entwickeln, basierend auf unserem Tool Codeling (<https://codeling.de>). Sie entwickeln Konzepte und Tools zum Vergleich der generierten Architektur mit einem normativen Architekturmodell und messen die Unterschiede.

Die Modellgenerierung und der Vergleich werden in eine Continuous Integration Pipeline integriert, so dass die Softwarearchitektur in ihrer implementierten und spezifizierten Form kontinuierlich verglichen werden kann.

Unser Framework Codeling¹ synchronisiert Modelle und Code. Dabei ist das Modell als eigenständiges Artefakt nicht mehr nötig. Codeling übersetzt Quelltextmuster in Modelle und zurück. Dies ermöglicht es, Modelle zuverlässig aus dem Quellcode zu extrahieren, um Änderungen an den extrahierten Modellen in den Code zu übertragen. Der generierte Code verfügt über klar definierte Schnittstellen zur Interaktion mit seinem kontextuellen Code, d.h. dem Rest des Systems. Für die Übersetzung zwischen Modellen und Code müssen Transformationen beschrieben werden, z.B. unter Verwendung von Java oder Xtend als Programmiersprachen.

Eines oder mehrere der folgenden Open-Source-Programme werden in dieser Arbeit betrachtet: JUnit 5, Neo4J, JGraLab, Your Favorite Open Source (Java) Program

In dieser Arbeit werden Sie die folgenden Fragen untersuchen:

- Was ist die Softwarearchitektur einer Reihe von Open-Source-Programmen?
- Wie wird die Softwarearchitektur als Modell beschrieben und wie wird sie implementiert?
- Wie kann ein Softwarearchitektur-Modell aus dem Code extrahiert werden?
- Wie kann man verwandte Softwarearchitekturen vergleichen und wie kann man ihre Überdeckung messen?

Für die Durchführung der Arbeit benötigte Kenntnisse: Java

Hilfreiches Wissen: Modellbasierte Softwareentwicklung

Motivation — English

Software architecture describes the most relevant design decisions for a software system on a high abstraction level. The description of the software architecture is a main driver for the product quality. Many software systems, including open source systems, do not have current architecture descriptions.

In this thesis you will analyze the source code and documentation of open source software systems for the underlying software architecture. You will develop tools for automatically generating software architecture models from the source code, based on our tool Codeling (<https://codeling.de>). You will develop concepts and tools for comparing the generated architecture with a normative architecture model and measure the differences.

¹<https://codeling.de>

The model generation and comparison will be integrated into a continuous integration pipeline, so that the software architecture as implemented and as specified can be compared continuously.

Our framework *Codeling*² synchronizes code by eliminating the need for models as a separate artifact. The tool translates models into source code patterns and back. This allows to reliably extract models from the source code to propagate changes in the extracted models to the code. The generated code has well-defined interfaces for interacting with its contextual code, i.e. the rest of the system. For translating between models and code, transformations need to be described, e.g. using Java or Xtend as programming languages.

One or more of the following open source programs will be considered in this thesis: JUnit 5, Neo4J, JGraLab, Your Favorite Open Source (Java) Program

In this thesis you will consider the following questions:

- What is the software architecture of a set of open source programs?
- How is the software architecture described as a model and how is it implemented?
- How can a software architecture model be extracted from the code?
- How to compare related software architectures and how to measure their compliance?

Knowledge requirements: Java

Helpful knowledge: Model-based software development

Organisatorisches

Kontakt:

Dr. Marco Konersmann <konersmann@uni-koblenz.de>

²<https://codeling.de>
