

Master / Bachelor
Inkrementelle Klonerkennung für Software-Modelle
Incremental Clone Detection for Software Models

Supervision and writing in English is welcome. For the English version of this announcement, see below.

Motivation

Um mit geringem Aufwand zu lauffähigen Systemen zu gelangen, wenden Softwareentwickler oft *copy und paste* an. Die so erzeugten Systeme sind jedoch oft anfällig für Programmierfehler und Sicherheitsprobleme, insbesondere, wenn kopierte Stellen unkritisch übernommen werden. Klonerkennung ist eine Technik, die es ermöglicht, vorhandene Klone in Code und Modellen systematisch zu erkennen und zu behandeln. Existierende Klonerkennungsverfahren sind jedoch oft sehr rechenintensiv. Eine kontinuierliche Behandlung von Klone ist dadurch in der Praxis erschwert.

Aufgabenstellung/Ziele

Ziel der Arbeit ist es, ein verbessertes Verfahren zur Erkennung von Klone in Softwaremodellen zu entwickeln. Gegenüber vorhandenen Verfahren aus der Literatur soll dieses eine vergleichbare Genauigkeit bei der Erkennung von Klone aufweisen. Bei mehrfacher Anwendung auf das gleiche Programm soll jedoch ein Laufzeitvorteil erreicht werden. Hierzu bietet es sich an, auf ein vorhandenes Verfahren aufzusetzen und dieses durch gezielte Speicherung und Verwendung von Zwischenergebnissen zu inkrementalisieren. Teilziele sind:

- Analyse vorhandener Klonerkennungsverfahren in Hinblick auf geeignete Einstiegspunkte für die Inkrementalisierung,
- Entwicklung einer geeigneten Repräsentation für Zwischenergebnisse,
- Entwicklung eines neuen/angepassten Klonerkennungsverfahrens unter Verwendung der Zwischenergebnisse und
- Evaluation dieses Verfahrens am Beispiel von Simulink-Modellen

Literatur

- ConQAT: State-of-the-Art-Klonerkennungstool für Simulink-Modelle
- Daniel Strüber, Jennifer Plöger, Vlad Acretoaie: Clone Detection for Graph-Based Model Transformation Languages. ICMT 2016: 191-206 (Best Paper Award ICMT2016)

Organisatorisches

Kontakt:
Dr. Daniel Strüber, Dipl.-Inf. (strueber@uni-koblenz.de)

Motivation

To minimize development effort during the creation of new systems, software developers often resort to *copy und paste* practices. The resulting systems are often prone to programming errors and security issues, in particular, if the copied parts are adopted without further critical assessment. Clone detection is a technology that supports the systematic identification and treatment of clones being present in a code base or model. However, the analysis strategies used by existing clone detection techniques are often computationally expensive. Therefore, the continuous treatment of clones in practice is complicated.

Objectives

The goal of this thesis is to develop an improved technique for the detection of clones in software models. In comparison to existing techniques from the literature, this technique should provide a comparable accuracy during clone detection. Yet, during the repeated application to the same model, a performance benefit should arise. To this end, a promising idea is to build on an existing technique, incrementalizing it by means of suitable storage and use of temporary results. Sub-goals are:

- investigation of existing clone detection techniques, focusing on suitable starting points for incrementalization,
- development of a suitable representation for temporary results,
- development of a new/adapted clone detection technique under use of temporary results, and
- evaluation of this technique by application to Simulink models.

Literature

- ConQAT: State-of-the-Art-Klonerkennungstool für Simulink-Modelle
- Daniel Strüber, Jennifer Plöger, Vlad Acretoaie: Clone Detection for Graph-Based Model Transformation Languages. ICMT 2016: 191-206 (Best Paper Award ICMT2016)

Contact

Dr. Daniel Strüber, Dipl.-Inf. (strueber@uni-koblenz.de)
