

MASTERARBEIT

SIMILARITY ANALYSIS OF COMPONENT AND CONNECTOR ARCHITECTURES

Ansprechpartner



Jörg Christian
Kirchhof, M.Sc. RWTH
Tel.: 0241 80 21311
kirchhof@se-rwth.de



David Schmalzing,
M.Sc. RWTH
Tel.: 0241 80 21343
schmalzing@se-rwth.de

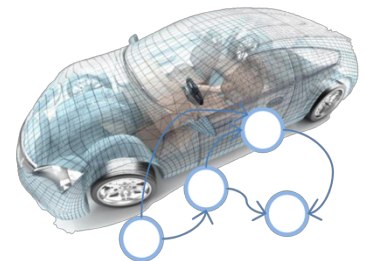
Aufgabenstellung

Prof. Dr.
Bernhard Rumpe
Tel.: 0241 80 21301
rumpe@se-rwth.de

Am Lehrstuhl für Software Engineering der RWTH Aachen ist ab sofort folgende studentische Arbeit zu vergeben

IHR AUFGABENUMFELD

Cyber-Physische Systeme müssen in der Lage sein, sich an ihre Umgebung anzupassen. Dazu müssen sie konfigurierbar sein. Software-Produktlinien erleichtern dies, indem gemeinsame Features der Varianten gemeinsam entwickelt werden und so zwischen den Varianten wiederverwendet werden können.



Wird eine Produktlinie weiterentwickelt, so ist es sinnvoll, bereits existierende Varianten zu aktualisieren. Das gleiche gilt, wenn sich die Feature-Auswahl, die in der existierenden Variante ausgeprägt ist, geringfügig ändert. Nach der Neugenerierung der Variante müssen alle Schritte, die sich auf diese Variante beziehen, z. B. Testdurchläufe, vom Nutzer gesichtet und ggf. wiederholt werden. Das ist sehr zeitaufwändig und damit teuer, jedoch in vielen Fällen unnötig, wenn sich nur kleine Teile der Variante geändert haben. Um diesen unnötigen Zusatzaufwand zu vermeiden, werden Analyseverfahren benötigt, mit denen Verhaltensäquivalenzen zwischen zwei Variantenversionen ermittelt werden können.

Ziel der Arbeit ist bestehende Ähnlichkeitsmetriken auf Komponenten-Konnektor Modelle anzuwenden und gegebenenfalls um weitere Analyseverfahren zu ergänzen. Die auf diese Weise für Komponenten-Konnektor Modelle bereitgestellten Analyseverfahren und Ähnlichkeits-metriken sollen anhand eines Fallbeispiels aus dem Bereich kooperative Fahrzeugautomatisierung evaluiert werden.

SCHWERPUNKTE

- Anpassung bestehender Verfahren auf neue Modelle
- Entwicklung von Softwareanalyseverfahren
- Evaluierung von Software Engineering Methodiken

WÜNSCHENSWERTE VORKENNTNISSE

- Java und Python Programmierung
- Modellbasierte Softwareentwicklung