Heiko Klare
IPD Reussner
Am Fasanengarten 5, 76131 Karlsruhe
heiko.klare@kit.edu

Professorinnen und Professoren

der KIT-Fakultät für Informatik

Karlsruhe, 07.10.2020

Vorstellung meines Promotionsthemas im Rahmen des Promotionsgesprächs (ehemals Professorenrunde)

Sehr geehrte Professorinnen, sehr geehrte Professoren,

ich beabsichtige, meine Promotionsprüfung an der KIT-Fakultät für Informatik des Karlsruher Instituts für Technologie abzulegen. Ich werde bei der Erstellung meiner Dissertation federführend von

Prof. Dr. Ralf Reussner

betreut. Meine Arbeit trägt derzeit den Arbeitstitel

Building Transformation Networks for Consistent Evolution of Multiple Models .

Ich möchte mich gerne bei Ihnen vorstellen und Ihnen das Thema meiner Arbeit erläutern. Eine kurze Zusammenfassung meiner Arbeit habe ich diesem Schreiben beigelegt.

Den wesentlichen Beitrag meiner Arbeit zur Erweiterung des Stands der Forschung in der Informatik fasse ich hiermit ganz kurz zusammen:

Die Entwicklung eines Software-Systems mittels verschiedener Artefakte erfordert Methoden für deren Konsistenzhaltung. Existierende Ansätze beschränken sich auf zwei Artefakte oder sind projektspezifisch. In meiner Dissertation untersuche ich systematisch, wie mehrere Spezifikationen zur Konsistenzhaltung von jeweils zwei Artefakten unabhängig entwickelt und modular gekoppelt werden können, sodass sie alle Artefakte konsistent halten. Ich stelle sowohl eine Formalisierung des Problems vor, als auch konstruktive und analytische Lösungsmethoden, deren Korrektheit ich beweise und deren Anwendbarkeit ich empirisch validiere.

Wenn Sie Interesse und Zeit für ein Gespräch mit mir haben, nennen Sie mir bitte mit beiliegendem Antwortschreiben einen möglichen Termin. Sollte ein Gespräch in nächster Zeit nicht möglich sein oder sollte Ihnen die schriftliche Zusammenfassung der Arbeit zunächst genügen, würde ich mich sehr freuen, wenn Sie mir dies mitteilen würden.

Mit freundlichem Gruß

Heiko Klare

Absenderdaten des Lehrstuhls

Heiko Klare
IPD Reussner, Geb. 50.34, Raum 328
Am Fasanengarten 5
76131 Karlsruhe

Ort, Datum

Rückantwort – Gespräch über Promotionsthema

☐ Ich möchte gerne mit Ihnen über Ihr Promotionsthema sprechen

Terminvorschlag: hier einen Terminvorschlag eintragen

☐ Von meiner Seite aus besteht derzeit kein Gesprächsbedarf.

Sonstige Mitteilung:

Hier kann der Lehrstuhl eine Nachricht an den/die Doktoranden/in eintragen

Mit freundlichem Gruß

Absendername

Karlsruher Institut für Technologie

Dissertationsvorhaben Heiko Klare

Erstgutachter: Prof. Dr. Ralf Reussner

Zweitgutachter: Prof. Dr. Colin Atkinson (Universität Mannheim)

Building Transformation Networks for Consistent Evolution of Multiple Models

In meiner Promotion habe ich die Konsistenzhaltung verschiedener Artefakte zur Beschreibung eines Softwaresystems untersucht, indem ich die Kopplung von Transformationen zwischen den Spezifikationssprachen dieser Artefakte formalisiert, analysiert und mit Methoden unterstützt habe.

Bei der Entwicklung eines Softwaresystems nutzen die verschiedenen Entwickler/-innen und weiteren Interessenvertreter/-innen diverse Sprachen zur Beschreibung unterschiedlicher Belange. Meist stellt der Programmcode das zentrale Artefakt dar, wird jedoch, implizit oder explizit, durch Spezifikationen der Architektur, des Deployments oder der Anforderungen ergänzt. Zur Spezifikation dieser Artefakte werden neben der Programmiersprache verschiedene andere Sprachen verwendet, beispielsweise UML für Architekturmodelle, OpenAPI für Schnittstellen-Spezifikationen oder Docker für Deployment-Spezifikationen von Microservices. Zur Erstellung eines funktionsfähigen Softwaresystems müssen diese Artefakte eine einheitliche, widerspruchsfreie Spezifikation des gesamten Systems darstellen. Beispielsweise müssen Service-Schnittstellen in all diesen Artefakt einheitlich repräsentiert sein. Wir sagen, die Artefakte müssen konsistent sein.

In der modellgetriebenen Entwicklung werden solch verschiedene Artefakte, dort allgemein *Modelle* genannt, bereits als wesentliche zentrale Entwicklungsbestandteile genutzt, um auch Teile des Programmcodes aus ihnen abzuleiten. Dies betrifft beispielsweise die Softwareentwicklung für Fahrzeuge oder allgemein die Entwicklung cyber-physikalischer Systeme. Zur Konsistenzhaltung der Modelle werden hier oftmals Transformationen eingesetzt, die nach Änderungen eines Modells die anderen Modelle anpassen. Die bisherige Forschung beschränkt sich jedoch auf die Konsistenzhaltung zweier Modelle und die aufeinander abgestimmte Konsistenzhaltung mehrerer Modelle. Ein systematischer Entwicklungsprozess, in dem einzelne Transformationen unabhängig und modular entwickelt und in verschiedenen Kontexten wiederverwendet werden können, wird hierdurch jedoch nicht unterstützt.

In meiner Dissertation befasse ich mich daher mit der Frage, wie Entwickler mehrere Transformationen zu einem Netzwerk kombinieren können, welches die Transformationen in einer geeigneten Reihenfolge ausführen kann, sodass abschließend alle Modelle konsistent zueinander sind. Dies geschieht unter der Annahme, dass einzelne Transformationen zwischen zwei Sprachen unabhängig voneinander entwickelt und daher nicht aufeinander abgestimmt werden. Meine Beiträge unterteilen sich dabei in die Untersuchung der Korrektheit einer solchen Kombination und die Auswirkung auf und Optimierung von Qualitätseigenschaften eines solchen Netzwerkes.

Ich leite zunächst einen adäquaten Korrektheitsbegriff her und definiere ihn mittels eines angemessenen Formalismus. Dieser Korrektheitsbegriff induziert die Notwendigkeit dreier weiterer Betrachtungen. Erstens fordert er von Transformationen eine *Synchronisations*-Eigenschaft, zweitens setzt er eine Form von *Kompatibilität* zwischen den Transformationen voraus, und drittens erfordert er das Finden einer korrekten Ausführungsreihenfolge der Transformationen, einer *Orchestrierung*. Ich leite her, wie mit bestehenden Sprachen Transformationen entwickelt werden können, die die Synchronisations-Eigenschaft erfüllen, und schlage auf Basis einer formal bewiesenen Eigenschaft ein entsprechendes Konstruktionsverfahren vor, dessen Anwendbarkeit ich in Fallstudien im Kontext der komponentenbasierten Softwareentwicklung evaluiere. Weiterhin definiere ich formal die

Karlsruher Institut für Technologie

Eigenschaft der Kompatibilität von Transformationen, für welche ich ein formales Analyseverfahren vorschlage und dessen Korrektheit ich beweise. Hieraus entwickele ich eine praktische Realisierung, deren Anwendbarkeit ich ebenfalls in Fallstudien nachweise. Außerdem untersuche ich, wie für Transformationen eine valide Orchestrierung, also eine Ausführungsreihenfolge nach der alle Modelle konsistent sind, gefunden werden kann. Dabei beweise ich, dass das allgemeine zugrundeliegende Problem unentscheidbar ist und stelle Indikatoren dafür vor, dass Einschränkungen des Problems, um dessen Entscheidbarkeit zu erreichen, die Anwendbarkeit unpraktikabel einschränken. Ich schlage daher einen Algorithmus vor, der die Lösung konservativ approximiert, und für den ich Korrektheit und eine wohldefinierte Eigenschaft zur Beschreibung seiner Nützlichkeit beweise. Des Weiteren kategorisiere ich Fehler, die auftreten können, wenn die identifizierten Korrektheitseigenschaften nicht erfüllt werden, und analysiere deren Relevanz in oben genannten Fallstudien. Hieraus leite ich zum einen ab, inwieweit das fehlerhafte Verhalten des Netzwerkes Rückschlüsse auf die Art des ursächlichen Fehler zulässt, und zeige zum anderen, dass sich die meisten potentiellen Fehler mit den vorgeschlagenen Verfahren per Konstruktion vermeiden lassen.

Zur Untersuchung von Qualitätseigenschaften eines Netzwerkes von Transformationen identifiziere ich zunächst relevante Eigenschaften und untersuche, wie sich verschiedene Typen von Netzwerktopologien auf diese auswirken. Hierbei zeigt sich, dass insbesondere Korrektheit und Modularität im Widerspruch stehen und die Wahl der Topologie ein Abwägen bei der Optimierung dieser Eigenschaften induziert. Ich leite hieraus ein Konstruktionsverfahren für Transformationsnetzwerke ab, welches die Notwendigkeit einer Abwägung zwischen den Qualitätseigenschaften abmildert und, unter gewissen Voraussetzungen, Korrektheit per Konstruktion gewährleistet. Für dieses Verfahren stelle ich außerdem eine Spezifikationssprache vor, die den Entwicklungsprozess strukturiert. Ich evaluiere die Anwendbarkeit des Verfahrens und der Sprache durch die Implementierung der bereits für die Korrektheitseigenschaften verwendeten Fallstudie aus der komponentenbasierten Softwareentwicklung.

Die Beiträge meiner Arbeit unterstützen sowohl Forscher/-innen, als auch Transformationsentwickler/-innen und Transformationsanwender/-innen bei der Analyse und Konstruktion von Netzwerken von Transformationen. Sie stellen für Forscher/-innen und Transformationsentwickler/-innen systematisches Wissen über die Korrektheit und weitere Qualitätseigenschaften solcher Netzwerke bereit, und zeigen insbesondere, welche Teile dieser Eigenschaften per Konstruktion oder per Analyse erreicht bzw. nachgewiesen werden können, sowie mit welchen Fehlern bei der Ausführung gerechnet werden muss. Zusätzlich stelle ich mit den Beiträgen konkrete, praktisch nutzbare Verfahren bereit, mit denen korrekte, modulare Netzwerke konstruiert, analysiert und ausgeführt werden können, wovon sowohl Transformationsentwickler/-innen als auch Transformationsanwender/-innen profitieren.