

Modellbasierte und Modellgetriebene Softwaremodernisierung

Workshop der Arbeitskreise „Modellgetriebene Software-Entwicklung“ (MDA), „Traceability/Evolution“ und „Langlebige Software-Systeme“ (L2S2) der GI-Fachgruppe „Architekturen“ im Rahmen der Konferenz „Modellierung 2014“

Steffen Becker

Universität Paderborn, Institut für Informatik
Zukunftsmeile 1, 33102 Paderborn
steffen.becker@upb.de

Matthias Riebisch

Universität Hamburg, Fachbereich Informatik
Vogt-Kölln-Str. 30, 22527 Hamburg
riebisch@informatik.uni-hamburg.de

Stefan Sauer

Universität Paderborn, s-lab – Software Quality Lab
Zukunftsmeile 1, 33102 Paderborn
sauer@s-lab.upb.de

Benjamin Klatt

FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe
klatt@fzi.ode

Thomas P. Ruhroth

TU Dortmund, Fakultät für Informatik
Otto-Hahn-Straße 14, 44221 Dortmund
thomas.ruhroth@cs.tu-dortmund.de

1 Motivation und Ziel des Workshops

Forderungen nach permanenter Änderbarkeit und nach verbesserten Qualitätseigenschaften von Softwaresystemen wie Performanz, Sicherheit und Zuverlässigkeit erfordern umfangreiche und regelmäßige Modernisierungsmaßnahmen. Diese Forderungen werden umso drängender, je wichtiger diese Systeme für Geschäftsprozesse und Produkte sind. Wegen der Kritikalität und der Größe der Systeme beinhaltet die Modernisierung sehr komplexe Aufgaben mit hohen Kosten und großen Risiken. Die Verwendung von Modellen kann helfen, die Komplexität zu beherrschen und durch frühzeitige Bewertung von Qualitätseigenschaften die Risiken zu verringern.

Im Fokus des Workshops stehen hierbei Modellierung und Visualisierung, Reverse Engineering und Refactoring sowie Qualität von Architekturen, außerdem die Transformation von Architekturmodellen sowie die Verbindung und Nachverfolgbarkeit von Architekturmodellierung mit Anforderungsbeschreibung und Implementierung. Modellbasierte Verfahren zur Integration und Migration sowie zur Prüfung und Sicherstellung der Konsistenz wie auch der Prüfung und Anpassung der Interoperabilität von Komponenten sind ebenfalls von Bedeutung.

Der Workshop dient der Identifikation neuer Forschungstrends und Herausforderungen auf Basis des aktuellen industriellen Bedarfs, der Suche nach geeigneten Ansätzen zur Problemlösung sowie zum Austausch von Erfahrungen mit modellbasierten und modellgetriebenen Techniken und Methoden für die Softwaremodernisierung.

2 Format und Ablauf des Workshops

Der 2. Workshop *Modellbasierte und Modellgetriebene Softwaremodernisierung* (MMSM 2014) fand am

19. März 2014 im Rahmen der Konferenz Modellierung 2014 in Wien statt. Er wurde gemeinsam von den Arbeitskreisen „Modellgetriebene Software-Entwicklung“ (MDA), „Traceability/Evolution“ und „Langlebige Software-Systeme“ (L2S2) der GI-Fachgruppe „Architekturen“ ausgerichtet. Mit mehr als 20 Teilnehmerinnen und Teilnehmern stieß der eintägige Workshop wie bei der ersten Auflage 2012 auf sehr große Resonanz. Das Programm des Workshops bot neben Vorträgen zu acht regulären Workshopbeiträgen und einem eingeladenen Vortrag von Uwe Zdun auch genügend Zeit für übergreifende Diskussionen am Ende jeder Session, die als Podiumsdiskussionen mit den Vortragenden organisiert waren. Die regulären Workshopbeiträge wurden im Vorfeld von einem 16-köpfigen Programmkomitee begutachtet und ausgewählt. Die acht ausgewählten Beiträge sind nachfolgend in dieser Ausgabe der Softwaretechnik-Trends zusammengestellt.

3 Eingeladener Vortrag

Der Workshop startete mit einem eingeladenen Vortrag von Uwe Zdun, Professor für Softwarearchitektur, der an der Universität Wien ein „Heimspiel“ hatte. In seinem Vortrag „Semi-automated Abstraction of Architectural Views Throughout the Software Lifecycle“ gab er einen sehr guten und passenden Einstieg zu vielen Fragestellungen, die im weiteren Verlauf des Workshops immer wieder aufgegriffen wurden.

Die Notwendigkeit der Software-Architecturevolution motivierte Uwe Zdun aus den bekannten Phänomenen des Abdriftens (Architecture Drift) und der Erosion. Als entscheidende Gegenmaßnahme schlug er die Rekonstruktion der Softwarearchitektur vor, wobei architektonische Abstraktionen und Sichten wesentliche Bestandteile der Rekonstruktion sind. Er diskutierte die Vor- und Nachteile existierender Bottom-up-, Top-

down- und hybrider Rekonstruktionsverfahren und leitete hieraus drei wesentliche Herausforderungen ab: (A) Abstraktionen auf verschiedenen Granularitätsebenen, (B) Traceability zwischen Architekturmodellen und Code sowie (C) Schritthalten mit der fortschreitenden Evolution der Software. Als Lösung stellte Zdun einen modellgetriebenen Ansatz vor. Mit einer Domain-Specific Language (DSL) werden Abbildungen zwischen Architektursichten spezifiziert, so dass z.B. aus Klassenmodellen automatisch Komponentenmodelle generiert und ebenfalls mittels automatischer Modelltransformationen Traceability-Links erstellt werden können. Mithilfe von Modellvergleichen können dann Unterschiede zwischen Versionen des Komponentenmodells, d.h. dem rekonstruierten und dem existierenden oder vorherigen Architekturmodell, identifiziert werden. Durch einfache Filter, die die Abstraktion unterstützen, und die Prüfung von Design Constraints können effektiv Abweichungen zwischen Architekturmodell und Code im Transformationsprozess entdeckt werden. In empirischen Untersuchungen konnte gezeigt werden, dass Traceability-Links eine wichtige Rolle für das Verstehen der architektonischen Abstraktionen spielen.

In der anschließenden Diskussion wurden mehrere aktuelle Forschungsthemen und Herausforderungen angesprochen: das Problem realistischer empirischer Studien zur Evaluation, die Rolle von Architekturwissen für Architekturentscheidungen, die Kombination von statischen Rekonstruktionsverfahren mit dynamischen Analysen sowie Softwarearchitektur-Metriken und Qualitätsannotationen der Architekturmodelle.

4 Vorträge zu regulären Beiträgen

Traceability, Transformationen und die Beziehungen zwischen Modellen und Code waren auch die Schwerpunktthemen des ersten Blocks regulärer Workshopbeiträge unter der Überschrift „Models to Code to Models“. Mahdi Derakhshanmanesh (Universität Koblenz-Landau) diskutierte in seinem Vortrag die Rolle von Modellen und Code, ihre Interaktion und Austauschbarkeit für die Entwicklung und Evolution von Softwaresystemen, bis hin zur Verwendung von Modellen zur Laufzeit. Einen Reverse-Engineering-Prozess zur Rekonstruktion von Traceability zwischen Code, Testfällen und Anforderungen, der statische und dynamische Code-Analyse kombiniert, stellte Harry Sneed (AN-ECON GmbH) vor. Der Beitrag von Firdaus Harun et al. (RWTH Aachen) stellte eine grundlegende Architektur für die flexible, toolbox-basierte Architekturrekonstruktion vor. Ziel des auf Instrumentierung beruhenden Ansatzes ist es, die Konsistenz zwischen beabsichtigter und realer Architektur u.a. mittels Architekturrekonstruktion und -überwachung, Traceability-Links, einheitlicher Terminologie sowie durch den Einsatz von Metriken zu wahren.

Unter dem Titel „Quality Aspects“ widmete sich der zweite Vortragsblock verschiedenen Ansätzen zur Ermittlung und Bewertung von Qualitätsaspekten. Wolf-

gang Goerigk (b+m Informatik AG) stellte einen modellgetriebenen Ansatz für die Entwicklung von Last- und Performanztests in DynaMod vor. Mit Hilfe dynamischer Analyse, automatischer Extraktion und manueller Spezifikation wird ein Workload-Modell konstruiert, aus dem der Workload für das getestete System generiert wird. Das Projekt Q-MIG wurde von Jan Jelschen (Universität Oldenburg) präsentiert. Kernidee ist die Kombination einer modellgetriebenen Softwaremigrationswerkzeugkette mit einem Qualitätscenter, in dem mit Metriken, Analysewerkzeugen und Qualitätsberichten das Qualitätsmanagement und zukünftig auch die Qualitätsvorhersage unterstützt werden sollen. Im Zentrum steht ein Qualitätsmodell für die Softwaremigration. Ein Quality-Smell-Katalog für Android-Entwickler wurde von Jan Reimann (TU Dresden) vorgestellt. Code wird hier als Modell betrachtet. Quality Smells sind Bad Smells, die eine gegebene Qualitätseigenschaft des Modells negativ beeinflussen. Durch die Anwendung zugeordneter Refactorings können diese Quality Smells aufgelöst werden.

Der letzte Vortragsblock befasste sich mit „Evolution and Migration“. Marvin Grieger (Universität Paderborn) et al. haben einen Restrukturierungsprozess für die Migration von Legacy-Systemen in eine serviceorientierte Architektur entwickelt. Der Prozess kombiniert zwei halbautomatische Clustering-Verfahren: hierarchisches und partitionierendes Clustering. Einen modellgetriebenen Ansatz für die kontrollierte Evolution von Code und verknüpften Modellen, der auf der Klassifikation von Code-Änderungen basiert, zeigte Michael Langhammer (KIT). Können Codeänderungen nicht automatisch klassifiziert werden, kann der Entwickler zusätzliche Informationen zur Intention der Änderung beisteuern. Mit einem Korrespondenzmodell wird Traceability und das Abdriften der Modelle vom Code überwacht.

5 Diskussion

In den Diskussionsrunden wurden außerdem folgende Themen übergreifend diskutiert: Evolution, Traceability, Refactoring, Qualitätseigenschaften auf Architekturebene, Metriken, explizites Architekturwissen zur Unterstützung von Architekturentscheidungen, Instrumentierung von Legacy-Code, Application Lifecycle Management und Nutzung von Software-Repositoryn. Ein weiteres Themengebiet waren aktuelle Trends in der modellbasierten und modellgetriebenen Softwaremodernisierung, u.a. textuelle vs. graphische Modellierungssprachen, Generierung vs. Interpretation.

6 Schlussbemerkungen

Wir bedanken uns herzlich bei den Autorinnen und Autoren, den Mitgliedern des Programmkomitees, den Teilnehmerinnen und Teilnehmern des Workshops sowie dem Organisationsteam der Modellierung 2014.

Weitere Informationen zum Workshop finden Sie unter http://akmda.ipd.kit.edu/mmsm/mmsm_2014/