

Musterbasiertes Re-Engineering von Softwaresystemen

Zusammenfassung

Software muss durch Re-Engineering kontinuierlich an eine veränderte Umwelt und neue Anforderungen angepasst werden, so dass ihre Wartbarkeit von entscheidender Bedeutung ist. Die vorliegende Arbeit stellt einen Ansatz vor, der das Re-Engineering von (objektorientierter) Software durch eine Beurteilung und Verbesserung ihrer Wartbarkeit auf Basis von Softwaremustern unterstützt.

Dazu wird ein Verfahren zur automatisierten Erkennung und Bewertung von Entwurfsmusterinstanzen im Quelltext einer Software für die Erkennung von Schwachstellen, Instanzen von Bad Smells und Anti Patterns, erweitert. Aufbauend darauf wird eine grafische Sprache zur Spezifikation von Programmtransformationen entwickelt, die Schwachstellen in bessere Lösungen transformieren.

Bei einer solchen Transformation soll i.d.R. die Struktur verbessert werden, ohne das von außen beobachtbare Verhalten zu verändern (Refactoring). Um bestimmte Verhaltensveränderungen auszuschließen, wird ein automatisches Verfahren entwickelt, dass eine Verifikation von Transformationen auf Einhaltung definierbarer Kriterien unterstützt. Die Kriterien beschreiben strukturelle Eigenschaften eines Programms, die durch eine Transformation erhalten oder vermieden werden müssen. Das Verfahren versucht nachzuweisen, dass die Kriterien (induktive) strukturelle Invarianten sind, die in allen Versionen beliebiger Programme gelten, die durch Ausführung vollständiger Transformationen entstehen können. Die Transformationssprache wurde dazu sowohl auf die Durchführung von Transformationen als auch ihre Verifikation ausgelegt.

Kann eine Transformation ein Kriterium verletzen, werden systematisch repräsentative Beispiele für problematische (Teil-)Ausführungen ermittelt und dem Re-Engineer so Hinweise für eine Korrektur gegeben.