

Die hohe Kunst der GUI-bezogenen Testautomatisierung

Eine stabile, flexible und plattformunabhängige GUI-Testautomatisierung – welcher QA-Manager träumt nicht davon? Aber muss das ein Traum bleiben? Nein! Bereits seit einiger Zeit ist mit web2test und QF-Test eine neue Generation von Testautomatisierungswerkzeugen am Markt verfügbar, die durch konsequente Weiterentwicklung diesen Traum bereits in die Realität umgesetzt hat.

Besondere Anforderungen des GUI-Testens

Die besondere Herausforderung beim Testen einer Software über die grafische Benutzerschnittstelle liegt in der Vielzahl der möglichen Zustände, die sowohl aktiv durch Nutzerinteraktionen als auch passiv durch die Software selbst bestimmt werden können. Dabei müssen beim Abspielen der Testskripte die richtigen GUI-Komponenten (Widgets) für die Wiedergabe der Interaktionen und Prüfpunkte identifiziert werden. Erschwert wird dies z. B. bei Weboberflächen zusätzlich durch den Trend hin zu einer dynamischen Oberflächengestaltung mittels AJAX/DHTML.

Auch ist die Weiterentwicklung von Software neben Funktionsänderungen oft mit Änderungen am Layout verbunden. Um die Wiederverwendbarkeit einmal erstellter Testskripte zu gewährleisten, sollten sie daher zu einem möglichst hohen Grad tolerant gegenüber diesen Änderungen sein.

Wie eingangs erwähnt, gibt es mit dem auf Web-Frontends spezialisierten web2test oder mit QF-Test für das Testen von Java und Web-Frontends bereits Testwerkzeuge, die mit ihrem komponentenbasierten Ansatz eine Antwort auf diese speziellen Herausforderungen des GUI-Testens haben.

Der komponentenbasierte Ansatz als Lösung

Warum ist gerade ein komponentenbasierter Ansatz die Antwort auf die oben genannten Herausforderungen? Zum besseren Verständnis soll dieser Ansatz am Beispiel von QF-Test und web2test näher erläutert und seine Vorteile aufgezeigt werden.

Technisch zeichnet sich der komponentenbasierte Ansatz dadurch aus, dass die Kommunikation der Testsoftware mit der zu testenden Anwendung direkt auf Ebene des jeweiligen Widget-Toolkits erfolgt – für Java über das API von Swing bzw. Eclipse/SWT und für Webanwendungen über das DOM. Dies bietet mehrere Vorteile:

Über die Testsoftware lassen sich problemlos systeminterne Komponenteninformationen und Werte auslesen, was den Umgang mit komplexen Komponenten wie Tabellen und Bäumen – auch wenn sie via AJAX implementiert sind – möglich macht. Zusätzlich erlaubt der Zugang zu diesen Komponenteninformationen detaillierte und vielseitige Tests wie z.B. die Sichtbarkeitsüberprüfung einer Baumansicht.

Da die Schnittstelle zum Betriebssystem beim komponentenbasierten Ansatz nur in Ausnahmefällen genutzt wird, schafft er zudem die optimale Voraussetzung für plattformunabhängige Tests. Neben der Plattformunabhängigkeit wurde bei QF-Test und web2test auch auf die Browserunab-

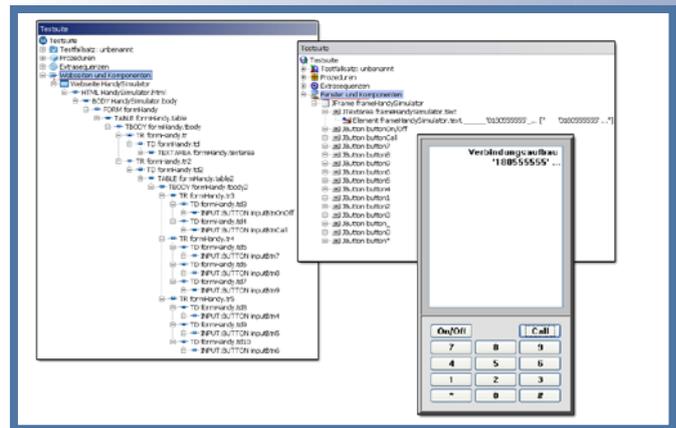


Abb. 1: Komponentenstruktur eines Handysimulators links für Web und rechts für Java

hängigkeit der Skripte beim Testen von Webanwendungen Wert gelegt.

Ein weiterer Vorteil ist die Stabilität der Testskripte. web2test und QF-Test nutzen komplexe Algorithmen zur intelligenten Wiedererkennung der verwendeten Komponenten. Dabei werden über die Schnittstelle zum Widget-Toolkit verfügbare Informationen, wie ID und Namen, aber auch hierarchische GUI-Struktur, Komponententitel, -beschriftung oder -geometrie ausgewertet. So lassen sich selbst nach Layoutänderungen der zu testenden Anwendung einmal aufgezeichnete Komponenten ohne Schwierigkeiten wieder finden. In Verbindung mit der echten Nutzersimulation von QF-Test und web2test beim Abspielen von Testskripten stellen auch Technologien wie AJAX und DHTML, mit ihrer Möglichkeit zur individuellen Oberflächengestaltung, kein Problem mehr dar.

web2test oder QF-Test?

web2test wurde als Kooperationsprojekt der itCampus Software- und Systemhaus GmbH und dem Softwareentwicklungshaus Quality First Software (QFS) entwickelt und ist eine Erweiterung des seit Jahren führenden Java GUI-Testtools QF-Test von QFS.

Warum wurde neben QF-Test mit web2test ein zweites eigenständiges Werkzeug auf den Markt gebracht? Ganz einfach: Während QF-Test mit Swing, SWT und Web drei verschiedene Widget-Toolkits unterstützt und dabei seinen Schwerpunkt weiterhin bei Java hat, wurde web2test speziell auf die Bedürfnisse des Webtestens abgestimmt. Die web-spezifischen Anpassungen, die schlankere Oberfläche und die aus Sicht des Webtestens überarbeitete umfangreiche Dokumentation machen es somit zum bevorzugten Werkzeug für Tester, die ihren besonderen Fokus auf dem Testen von Webanwendungen haben, während QF-Test das ideale Tool für Anwender mit Java Projekten bleibt und Webtesten als Zusatzfunktion bietet.

Der Autor - **Benjamin Franke** - (B.Franke@itcampus.de) ist verantwortlich für das Product Marketing von web2test, der neuen Web-Testsoftware von itCampus.
www.web2test.de