

# Eine Testautomatisierung für Roboter-Steuerprogramme auf Android unter Verwendung von TTCN-3

Yassine El Amrani, Hans W. Nissen

Fachhochschule Köln  
Fakultät 07, Institut für Nachrichtentechnik  
Betzdorfer Str. 2, 50679 Köln  
yassine.elamrani@gmail.com, hans.nissen@fh-koeln.de

**Abstract** Mit Hilfe von Programmen auf Smartphones kann man immer mehr Geräte wie Roboter oder Helikopter steuern. Das Überprüfen des korrekten Verhaltens derartiger Programme ist aufwendig und zumeist ein manueller Vorgang. Wir haben eine auf TTCN-3 basierende Testumgebung erstellt, die die automatisierte Prüfung von Android-Steuerprogrammen ermöglicht. Ein Testfall kann vollständig automatisiert ablaufen, wobei auch die Benutzeroberfläche auf dem Smartphone bedient werden kann. Ein Tester kann über ein Kontrollfenster auch leicht einen ad-hoc-Testfall erstellen, indem er dort die Sensor- und Aktuatorwerte dieses speziellen Testfalls definiert.

## 1 Einleitung

Schon seit einiger Zeit werden erschwingliche Hobby-Hubschrauber, Quadcopter, Roboter (z.B. Lego Mindstorms) u.v.a. zusammen mit Steuerprogrammen für Smartphones (Android oder iOS) angeboten. Diese eher spielerischen Anwendungen sind von ihren Prinzipien jedoch gar nicht weit entfernt von ernsthaften Steuerprogrammen für eingebettete Systeme.

Es existieren bereits Werkzeuge, die den Unit-Test von Android-Programmen unterstützen [ATF]. Durch die Verwendung von JUnit und Mock-Objekten kann die funktionale Korrektheit auf Modulebene geprüft werden. Ein umfassender Systemtest eines Steuerprogramms auf Android erfordert jedoch die Interaktion mit der zu steuernden Hardware. Es sind verschiedene und teilweise komplizierte Testaufbauten für die Roboter und ihrer Umgebung erforderlich. Insbesondere die Prüfung von Extremsituationen (z.B. Hält der Roboter an der Tischkante an?) kann hierbei aufwendig und problematisch sein.

Die Testing and Test Control Notation Version 3 (TTCN-3) [WDTK11] ist eine etablierte und verbreitete Technologie zur Software-Prüfung in der Telekommunikation, im Automobil-Sektor und vielen weiteren Branchen, in denen ein Gesamtsystem aus verteilten, kommunizierenden Komponenten aufgebaut ist. Deshalb haben wir uns entschieden, TTCN-3 als Grundlage für unsere automatisierte Testumgebung zu verwenden.

## 2 Konzept der Testumgebung

Als Inspiration für unsere Testumgebung dient ein Flugsimulator zur Ausbildung von Piloten: Der Pilot sitzt hierbei in einem nachgebauten, beweglichen Cockpit mit allen Instrumenten und einer visuellen Präsentation der Aussicht aus dem Cockpit. Man kann grob zwei unterschiedliche Arten zur Durchführung einer Simulationssitzung unterscheiden: *Vordefiniertes Szenario*: Es wird dem Piloten ein genau definiertes Szenario (z.B. Landeanflug auf Frankfurt bei schlechtem Wetter) präsentiert, auf das er korrekt reagieren muss. *Ad-hoc Szenario*: Der Ausbilder nimmt über eine Kontrolleinheit bewusst Einfluss auf die Eigenschaften des simulierten Flugzeugs und der Umwelt (z.B. Ausfall von Instrumenten, Vorgabe ungewöhnlicher Messwerte). Hierdurch können Sonderfälle und Extremsituationen individuell simuliert und geprüft werden.

In unserer Situation entspricht das Roboter-Steuerprogramm auf dem Smartphone dem Piloten, denn dieses analysiert die gemessenen Sensorwerte und gibt daraufhin Steueranweisungen an die Aktuatoren. Das TTCN-3 Testsystem entspricht dem Flugsimulator. Es simuliert das zu steuernde Gerät und nimmt die Steuerbefehle an die Aktuatoren entgegen und bewertet diese.

## 3 Umsetzung der Testumgebung

Abbildung 1 zeigt alle Komponenten im gesamten System. Die grau gefärbten Komponenten sind welche die in diesem Projekt neu entwickelt wurden. Die anderen Module sind Module, die im System angewendet und integriert wurden (z.B. die Robosmart Middleware).

**teil-automatisierte Durchführung:** Für die teil-automatisierte Durchführung ist die traditionelle TTCN-3 Anpassung an das System Under Test (SUT) erforderlich: System Adapter und Codec. Die Kommunikation zwischen dem Steuerprogramm und dem Lego Mindstorms Roboter erfolgt durch das LEGO Communication Protocol (LCP). Daher war die erste Aufgabe in diesem Projekt einen TTCN-3 LCP Adapter zu entwickeln. Diese Aufgabe wurde in einem speziellen Codec (*CD/TCI*) implementiert. Das Steuerprogramm nutzt Bluetooth als physikalischen Kommunikationsweg zu dem Roboter. Daher wurde ein

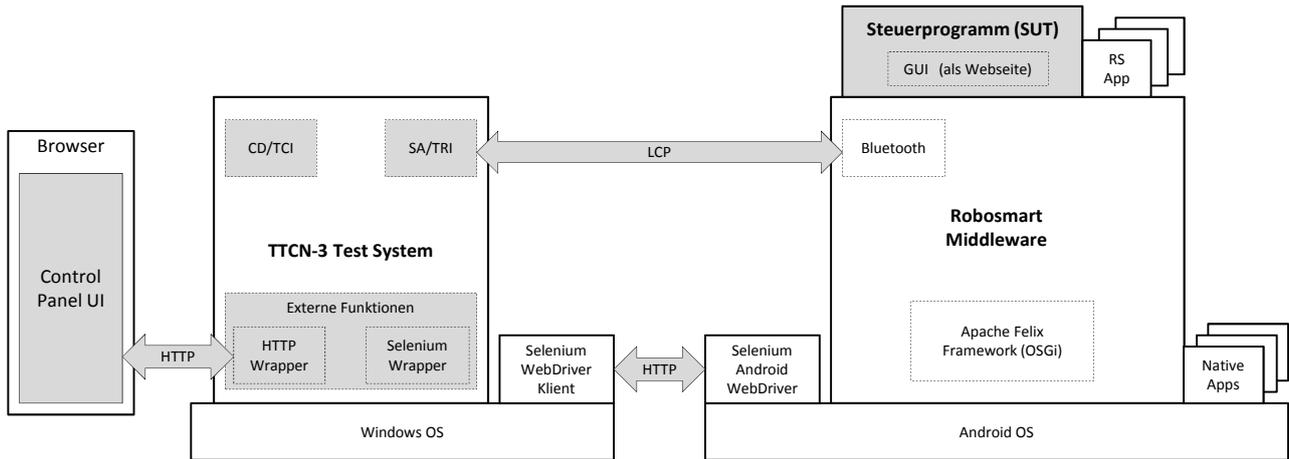


Abbildung 1: System-Architektur

TTCN-3 Bluetooth System Adapter (*SA/TRI*) entwickelt. Dafür wurde die Schnittstelle der Bibliothek *BlueCove* eingesetzt.

#### vollständig automatisierte Durchführung:

Zur automatischen Bedienung der GUI des Steuerprogramms aus dem TTCN-3 Test-Modul wurde das Selenium WebDriver Framework [Selen] angewendet. Das Framework ist für das automatisierte Testen von Web-Applikationen entwickelt worden. Für Android Mobilgeräte gibt es einen entsprechenden Treiber, der auf dem Android Smartphone installiert wurde (*Selenium Android WebDriver*). Der Selenium WebDriver wartet auf Anfragen eines Selenium WebDriver Klienten. Ein derartiger Klient befindet sich auf dem Rechner mit dem TTCN-3 System (*Selenium WebDriver Klient*). Dieser Klient stellt eine Schnittstelle zur Verfügung, um HTTP-Anfragen zu erstellen, die den Server zu entsprechenden Aktionen auf der Benutzeroberfläche des Steuerprogramms veranlassen. Um diese Schnittstelle aus den TTCN-3 Testprogramm nutzen zu können, wurde ein Wrapper (*Selenium Wrapper*) entwickelt und als externe Funktionen in das TTCN-3 Testsystem integriert.

**Durchführung mit ad-hoc-Testfall:** Das Kontrollfenster, in dem die Eingaben für einen ad-hoc-Testfall erfolgen, ist als Webseite realisiert (*Control Panel UI*) (siehe Abbildung 2). Auf ihr können die simulierten Werte für Sensoren (linke Seite des Fensters ("Input")) und die erwarteten Steuerbefehle für Aktuatoren eingestellt werden (rechte Seite des Fensters ("Output")). Zur Übermittlung der eingestellten Werte stellt der *HTTPWrapper* einen HTTP-Socket und Schnittstellen-Operationen bereit, um die eingestellten Sensor- und Aktuator-Werte auszulesen. Diese Operationen sind als externe Funktionen in das TTCN-3 Testsystem eingebunden. Innerhalb eines Testfalls kann somit auf die eingestellten Werte zugegriffen werden.

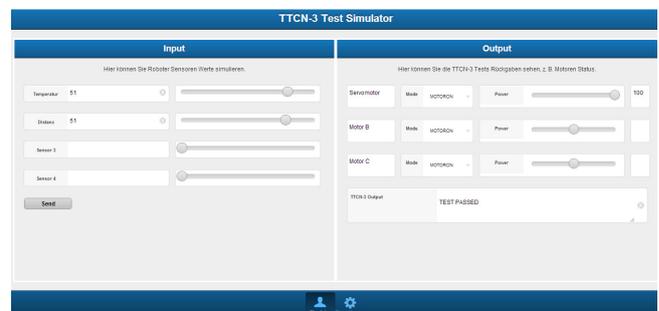


Abbildung 2: Kontrollfenster für ad-hoc Testfall

## 4 Fazit und Ausblick

Die hier beschriebene Testumgebung ist das Ergebnis der Masterarbeit von Herrn El Amrani. Die Umgebung ermöglicht das automatisierte Testen von Android-Steuerprogrammen. Die beiden wesentlichen Ergebnisse sind (a) die Integration der Bedienung der Benutzeroberfläche des Steuerprogramms in einen vollständig automatisierten TTCN-3-Testfall, der z.B. für den Regressionstest verwendet werden kann und (b) das Ermöglichen von ad-hoc TTCN-3-Testfällen durch das Setzen von Testwerten über ein externes Kontrollfenster.

Als Erweiterung sollte ein ad-hoc Testlauf nicht nur das einmalige Ausführen eines Testfalls umfassen, sondern es sollte eine kontinuierliche Testausführung ermöglicht werden. Weiterhin planen wir die Entwicklung eines einfachen Capture-Replay-Werkzeugs.

## Literatur

- [ATF] Android Testing Framework, <http://developer.android.com/tools/testing>.
- [Selen] SeleniumHQ Browser Automation: WebDriver, <http://docs.seleniumhq.org/projects/webdriver/>.
- [WDTK11] C. Willcock, T. Deiß, S. Tobies, S. Keil, F. Engler, S. Schulz: An Introduction to TTCN-3, John Wiley & Sons, 2. Auflage, 2011.