

# Testen mit TTCN-3 in der Praxis

Andrej Pietschker

Siemens AG

Corporate Technology, Software & Engineering

Otto-Hahn-Ring 6, 81730 München, Germany

andrej.pietschker@siemens.com

**Einleitung.** Die Test und Test Control Notation (TTCN-3) ist eine neue Testdesign- und -implementierungssprache, die viele Varianten von Black-Box-Tests unterstützt. Insbesondere eignet sie sich zum Test von reaktiven Systemen, ganz gleich, ob es sich um sequentielle, parallele oder verteilte Systeme handelt.

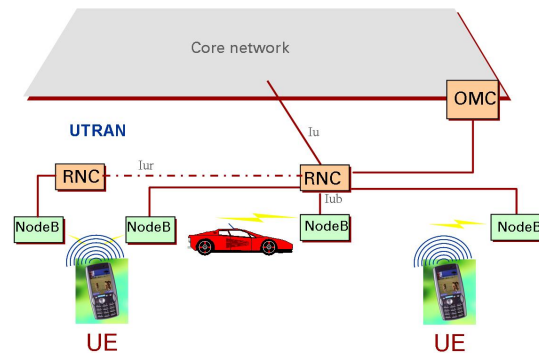
Frühere Versionen von TTCN wurden hauptsächlich in der Telekommunikationsindustrie angewendet. Die neue Version ist universeller und kann deshalb in vielen weiteren Gebieten eingesetzt werden. Nicht alle möglichen Einsatzgebiete sind bisher untersucht worden, aber immer mehr Interessierte aus Industrie und Forschung arbeiten mit TTCN-3 (siehe <http://www.ttcn-3.org/>).

TTCN-3 ist die einzige standardisierte Testbeschreibungssprache der Welt. Für Unternehmen ist die Möglichkeit, spezialisierte Testsysteme durch ein standardisiertes Werkzeug zu ersetzen, eine sehr interessante Alternative. In unserem Vortrag behandeln wir zwei Beispiele für den Einsatz von TTCN-3 und zeigen auf, was wir dabei gelernt und welche Schlüsse wir gezogen haben.

**Testautomatisierung — Warum?** Iterative Entwicklung mit kurzen Entwicklungszyklen und täglichen Builds wird als die Antwort auf die Probleme in der traditionellen Entwicklungsmethodik gesehen. Diese muß jedoch in Frage gestellt werden, wenn keine automatische Verifikation möglich ist. Obwohl Testautomatisierung eine Investition darstellt, werden Zeit und Kosten eingespart, da Softwarefehler in frühen Phasen und zeitnah zur Implementierung gefunden werden. Auch die Entwicklung profitiert von der Testautomatisierung, denn sie ermöglicht Verifikation nach einem Refactoring. Zusammenfassend kann man sagen, daß automatisiertes Testen eine Voraussetzung für iterative Softwareentwicklung ist.

**Einsatz von TTCN-3 in entwicklungsnahen Tests.** TTCN-3 ist für den Einsatz in entwicklungsnahen Tests geeignet. Als Beispiel gehen wir auf die Entwicklung einer TTCN-3-basierten Testsuite im

UMTS-Bereich ein. Folgendes Bild hilft bei der Einordnung des Produktes.



Ein NodeB übernimmt in einem UMTS-Netzwerk die Kommunikation mit UMTS-fähigen Endgeräten auf der einen Seite und stellt auf der anderen Seite die Verbindung zum Radio Network Controller (RNC) her.

In der Vergangenheit hat man TTCN-2 zur Testautomatisierung eingesetzt. In unserem Projekt wurde untersucht, welche Vorteile und Hindernisse sich beim Umstieg von TTCN-2 auf TTCN-3 bieten.

Ein klarer Vorteil von TTCN-3 gegenüber TTCN-2 ist die moderne Notation. Mit TTCN-3 können Tests von jedem, der eine Programmiersprache beherrscht, gelesen (und geschrieben) werden. Es ist im Gegensatz zu TTCN-2 kein spezieller Editor nötig. Emacs oder ein beliebiger anderer Editor sind ausreichend. Dadurch fallen geringere Kosten bei Werkzeugen an. Die Testfallimplementierung ist vom ausgewählten Werkzeug unabhängig. Das ASCII-Format vereinfacht Nutzung von textbasierten Werkzeugen wie z.B. diff/merge. Wichtig ist auch, daß die Nutzung neuester Technologien eine zusätzliche Motivation für die Beteiligten bietet.

Neue Sprachelemente wie z.B. blockierende Kommunikation oder dynamische Konfiguration und Ausführung mittels create, start, connect, map ermöglichen die Implementierung von Testfällen, die sich bisher schwer oder garnicht automatisieren ließen.

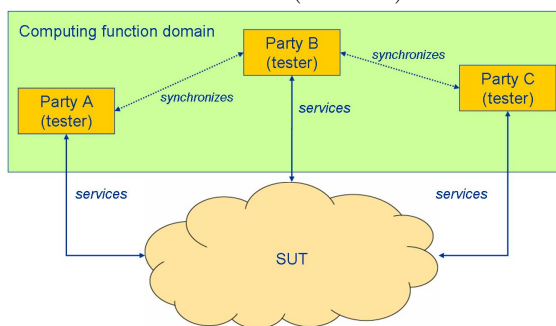
Standardisierte Schnittstellen bei TTCN-3 zur Anbindung des System unter Test und des Testmana-

gements führen zu weitgehender Unabhängigkeit vom Werkzeughersteller.

Die Modularisierung bietet weitere Vorteile, wie z.B. einen einfacheren Umgang mit Versionskontrollwerkzeugen. Testsuite-Design vereinfacht die Arbeit im Team. Große Projekte können aufgeteilt werden, nur die geänderten Dateien müssen neu übersetzt werden.

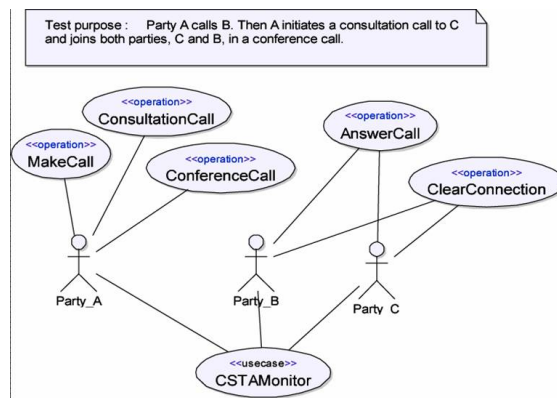
Wir haben gelernt, daß Kenntnisse beider Sprachen notwendig sind, um den Umstieg zu schaffen. Das Testdesign ist bei TTCN-2 und TTCN-3 unterschiedlich, d.h. eine reine Konvertierung erzeugt eine TTCN-3-Testsuite im TTCN-2 Design. In diesem Projekt ist eine vollautomatische Konvertierung nicht möglich. Eine Kombination aus automatischer Konvertierung und manueller Nacharbeit ist unumgänglich. Der Umstieg muß geplant und ein günstiger Zeitpunkt dafür gewählt werden. Die Umstellung sollte nicht unter Zeitdruck durchgeführt werden.

**Einsatz von TTCN-3 in Systemtests.** Dieses Beispiel zeigt die Möglichkeiten des Einsatzes von TTCN-3 außerhalb der Welt von Protokolltests auf. Es werden Endgeräte in TTCN-3 simuliert, die in dem System unter Test Dienste (Services) aufrufen.

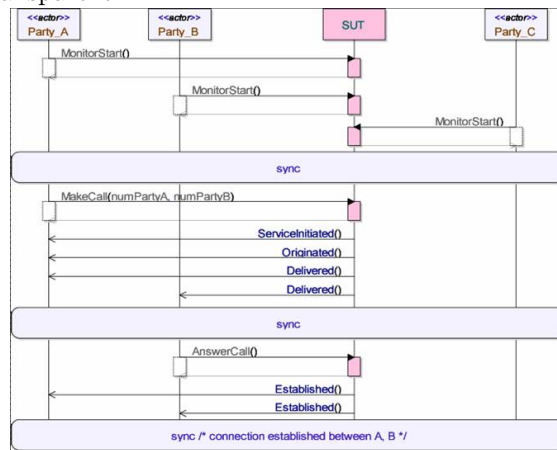


In diesem Projekt wurde eine Architektur spezifiziert und implementiert. Durch ein Bibliothekskonzept wurden das Aufsetzen der Konfiguration automatisiert und sich wiederholende Schritte in Funktionen verborgen. Dadurch wurde erreicht, daß sich die Testfälle nur in einem TTCN-3-Modul unterscheiden. Die Testfälle selbst stellen die Interaktion der simulierten Endgeräte dar. Dieses Verhalten wird in diesem einen Modul implementiert. Um die Spezifikation durch Nicht-TTCN-3-Experten zu erleichtern, wurden Elemente aus UML eingesetzt.

Die Spezifikation der Testfälle erfolgt mit Use-Cases und Message Sequence Charts (MSC). Im Use-Case sind die beteiligten Teilnehmer spezifiziert und die Services, die genutzt werden.



Das MSC stellt die Abfolge der Ereignisse im Testfall dar. Wichtig ist ein Konzept der Synchronisation der Teilnehmer untereinander, da sie prinzipiell unabhängig agieren. Auch dieses Konzept wurde als Bibliothek implementiert und ist für den Spezifizierer transparent.



Diese Abfolge wird dann in ein TTCN-3-Modul übersetzt und mit dem Framework zusammen als ausführbarer Testfall bereitgestellt.

Es bleibt zu vermerken, daß TTCN-3 auch im End-to-End Test erfolgreich einsetzbar ist. Eine gute Architektur und ein durchdachtes Bibliothekskonzept vereinfachen die Implementierung. Werkzeuge zum grafischen Testdesign ermöglichen auch Nicht-TTCN-3-Experten die Spezifikation von Tests. Eine auf dem UML 2.0 Test Profile basierende Lösung wird als zukunftsicher betrachtet.

Wichtig beim Aufsetzen des TTCN-3-Projekts ist, daß es wie ein normales Software-Entwicklungsprojekt behandelt wird. Konfigurationsmanagement, Testfallmanagement sowie Mitarbeiterschulungen müssen berücksichtigt werden.

**Rollen in einem TTCN-3 Projekt.** Organisationen, die Testautomatisierung bisher nicht in großem Umfang betrieben haben, müssen sich auf Veränderungen einstellen.

Die folgende Tabelle zeigt mögliche Rollen und die Werkzeuge, die genutzt werden.

Testdesigner (Programmierer)	Entwickelt Testsuiten	TTCN-3-Tool, Testdesign- Tool, Konfigu- rationsmanage- ment
Testdesigner (Systemtester)	Entwickelt Testsuiten grafisch	Testdesign- Tool, Konfigu- rationsmanage- ment
Testausführer	Führt Testsui- ten aus	TTCN-3- Tool zur Ausführung, Web-Browser
Testplattform- Programmierer	Entwickelt Ad- apter für die Anbindung der SUT	IDE, Konfigu- rationsmanage- ment
Testmanager	Benötigt Testreports und weitere Metriken	Testmanagement- Tool, Web- Browser

Diese Tabelle veranschaulicht, daß neue Rollen in traditionelle Systemtestorganisationen Einzug halten werden. Organisationen müssen sich darauf vorbereiten z.B. durch Mitarbeiterschulungen um die notwendigen Fähigkeiten aufzubauen.

**Zusammenfassung.** TTCN-3 unterstützt unterschiedliche Anwendungsszenarien. Je nach Einzelfall sprechen unterschiedliche Aspekte für den Einsatz von TTCN-3 als Testautomatisierungslösung. Wichtig ist die grundlegende Erkenntnis, daß Testautomatisierung für die iterative Entwicklung eine wesentliche Voraussetzung ist. Bei der Entwicklung einer Testlösung mit TTCN-3 ist es wichtig, die Regeln für erfolgreiche Softwareentwicklung zu beachten. Testautomatisierung muß geplant, eine Architektur entwickelt, Testfälle müssen spezifiziert und implementiert werden. Diese Schritte sind durch Projektmanagement und Versionskontrolle zu unterstützen.