Testautomatisierung für den Systemtest: Kosten, Nutzen, Möglichkeiten

Jörn Münzel
Bosch Telecom GmbH, 60277 Frankfurt/M
Tel.: 069/7505-6321, Email: joern.muenzel@fr.bosch.de

Der Systemtest, d.h. der funktionale Test von in Hardware eingebundener komplexer Software, ist oft eine sehr zeitintensive und teure Aufgabe. Bosch Telecom als Hersteller von u.a. großen privaten Kommunikationsnetzen hat darum das Projekt RESTATE (**RE**use of **S**ystem **T**est cases through **A**pplying a **T**TCN **E**nvironment, ESSI-PIE No. 23978) zur Bewertung des Einsatzes von TTCN im Bereich des System- und Regressionstests von Kommunikationssystemen durchgeführt.

TTCN (Tree and Tabular Combined Notation - ISO IS 9646) ist eine normierte Spezifikationssprache zur formalen Beschreibung von Testabläufen. Für den Einsatz von TTCN stehen eine Reihe von käuflichen Werkzeugen (Editoren, Compiler, Analyse- und Ausführungssupport) zur Verfügung, die einen hohen Automatisierungsgrad der Testausführung und -analyse unterstützen.

Der folgende Bericht beschreibt knapp die Struktur und Ziele des Projektes, die ermittelten Kosten der Testautomatisierung sowie den Nutzen und zukünftige Möglichkeiten.

Zielsetzung

Die Zielsetzung des Projektes RESTATE war die Reduzierung des Testaufwandes und der Testdauer im Bereich des funktionalen Systemtests durch werkzeuggestützte Testprogrammierung, -ausführung und -analyse. Dies galt insbesondere unter der Annahme, daß die heutigen und zukünftigen Entwicklungen von Kommunikationssystemen iterative Weiterentwicklung bereits bestehender Systeme sind. D.h. die Integration von neuen Funktionen und der Nachweis der Stabilität bestehender Leistungen ist eine Hauptaufgabe des funktionalen Systemtests. Das größte Problem in unserem Umfeld sind dabei die Vielzahl heterogener Testpunkte am System und die Notwendigkeit ihrer automatisierbaren Bedienung und ihrer Synchronisierung (z.B. Test einer Telefonverbindung zwischen unterschiedlichen Telefonterminals inklusive Teilnehmereinrichtung und Gebührenerfassung).

TTCN wurde als Basistechnologie für das Experiment ausgewählt, da es die einzige standardisierte Notation im Bereich des Testens von Telekommunikationsprodukten ist, die einerseits alle notwendigen Konstrukte und Basiskonzepte beinhaltet und andererseits bereits seit geraumer Zeit von Werkzeugherstellern unterstützt wird. Kern von TTCN ist das **PCO-Konzept** (Point of Control and Observation), mittels dessen das erwartete Verhalten eines Testobjektes an einem oder mehreren Punkten bzw. Schnittstellen durch Nachrichtensequenzen *programmiert* wird.

Projektstruktur

Im Projekt wurden zwei Testfallgruppen (Test Suites) in TTCN programmiert und in einer realen Umgebung eingesetzt. Vorbereitend dazu wurden die beteiligten Mitarbeiter geschult, die notwendigen Werkzeuge evaluiert, installiert und erprobt sowie grundlegende Konzepte zum Einsatz von TTCN im konkreten Umfeld erarbeitet (Prozeß, Design-/Programmierregeln).

Parallel dazu wurde die Durchführung derselben zwei Test Suites nach bisherigen Vorgehen aufwandsmässig erfaßt. Basis dieses Vorgehens ist die manuelle Stimulierung der Vermittlungsanlage aus Sicht der Benutzer mittels Endgeräten und die Beobachtung der Reaktion an unterschiedlichen Punkten (Protokoll-Traces, Hörtöne, Sprachkanäle, Displayanzeigen, etc.). Zur Automatisierung des Regressiontests können die Abläufe mittels eines Protokoll-Monitors aufgezeichnet und wieder abgespielt werden. Das Verfahren ist aber manuell aufwendig und die Aufzeichnungen sind nur bedingt wartbar bzw. wiederverwendbar.

Kosten

Die Kosten beim Aufbau einer *programmierbaren* Testumgebung lassen sich in die drei Bereiche Werkzeuginvestitionen, Schulung und Initialisierung untergliedern.

Die **Werkzeuginvestitionen** umfassen in unserem Fall die Beschaffung eines TTCN-Editors (ca. 8000,- DM je Unix-Lizenz) und einer TTCN-Ausführungsumgebung (Compiler, Execution, Analyse-Support, ca. 45.000,- DM je Lizenz). Die Ausführungsumgebung umfaßt nur Software-Kosten, da die Hardware (Protokolll-Simulator) vorhanden war. Nach unseren Nutzungsprofilen sollte je Tester ein Editor und je 3-5 Tester eine Ausführungsumgebung vorhanden sein.

Die **Schulungskosten** betrugen je Tester ca. 2.000,- DM für einen TTCN-Kurs und ca. 3.500,- DM für eine Schulung der TTCN-Werkzeuge im Bereich der Testumgebung. Unterstützend wurden die Tester 3 Monate durch einen erfahrenen TTCN-

Experten betreut. Eine Schulung der Tester im Bereich der Testdomäne war zusätzlich erforderlich und wurde intern von erfahrenen Testern geleistet.

Die **Initialisierungskosten** umfassen die Kosten, die notwendig waren eine Testarchitektur für unser System zu entwerfen und mit entsprechenden Designregeln, Prozeßschritten und TTCN-Konventionen zu unterstützen. Zusätzlich wurden erste prototypische Tests programmiert und eine Basisbibliothek von wiederverwendbaren Teilmakros (Test Steps) erstellt. Der Aufwand betrug ca. 8 Personenmonate.

Als Ergebnis der Kostenanalyse ist festzuhalten, daß so ein Testarbeitsplatz dem eines Software-Entwicklers vergleichbar ist. Dies betrifft auch die Voraussetzungen der Fähigkeiten für einen Tester, der Kompetenzen in Design, Kodierung (Programmiersprache) und Programmverifikation (Test) benötigt.

Nutzen

Einen direkten Nutzen aus dem Einsatz von TTCN und der damit möglichen Testautomatisierung läßt sich aus unseren Messungen nur bedingt ableiten. Wir haben aber vor allem erkannt, welche Bedingungen notwendig sind, Testautomatisierung nutzbringend einzusetzen.

Unsere Aufwandsmessungen haben ergeben, daß bei unserem bisherigen Testvorgehen ein Aufwand von ca. 3 - 6 Stunden je Testfall benötigt wird. Es besteht dabei eine starke Abhängigkeit zwischen der Komplexität des Tests (Anzahl Testpunkte und Testschritte) und des benötigten Aufwandes. Beim Einsatz von TTCN ergaben sich Durchschnittswerte von 7,2 bzw. 20 Stunden je Testfall, wobei hier eine starke Veränderung über die Zeit gemessen wurden. Während für die ersten Testfälle Aufwände bis zu 100 Stunden notwendig waren, konnten am Ende Tests in weniger als 3 Stunden realisiert werden. Dies war durch einen hohen Grad von Wiederverwendung möglich und zeigte auch kaum merkliche Abhängigkeiten von der Komplexität des Testfalles. Das Messen von Wartungsaufwänden konnte bisher nur bedingt durchgeführt werden, zeigt aber mögliche Aufwandsreduktionen von 70 - 100 % im Vergleich zum bisherigen Verfahren.

Nach ersten Berechnungen ergibt sich ein kostenmässiger Nutzen von TTCN somit erst bei einer 5-10 maligen Wiederverwendung der Testprogramme. Eine Reduktion der Testdauer kann früher erreicht werden, da die aufwendige Testprogrammierung parallel zur Entwicklung stattfinden kann. Dies im Gegensatz zur Aufzeichnung, die ein ablauffähiges System voraussetzt.

Bedingungen den Programmierungsaufwand zu reduzieren sehen wir in der Realisierung von adäquaten PCOs und in der engen Integration von Entwicklung und Testprogrammierung. Ein adäquater PCO ist in unserem Sinne eine automatisiert zugreifbare Systemschnittstelle. Diese sollte aufgabenbezogene Steuerungs- und Bewertungsmöglichkeiten haben sowie über die Zeit syntaktisch und semantisch stabil bleiben. Ein integratives Arbeiten der Entwickler und Tester an solchen Schnittstellen unterstützt einerseits die Tester stark im Verständnis der Aufgaben bzw. der Abläufe und unterstützt andererseits die Entwickler durch eine frühzeitige Prüfung ihre Entwürfe bzgl. Vollständigkeit und Konsistenz im System.

Möglichkeiten

Neben den erwähnten Möglichkeiten, Testpunkte und Tester möglichst frühzeitig in den Entwurf und die Entwicklung von Systemen zu integrieren, sehen wir für die Testautomatisierung mittels TTCN noch weitere Einsatzbereiche.

TTCN ermöglicht mittels (verfügbarer) Compiler die Generierung von ausführbaren Code für unterschiedliche Umgebungen, z.B. Wiederverwendung von Tests in Entwicklungs- **und** Zielumgebungen.

Weiterhin existieren bereits Werkzeuge TTCN aus formalen Entwurfsspezifikationen (SDL, MSC) zu generieren. SDL (Specification Description Language, ISO-Standard) unterstützt die formale Spezifikation von zustandsbasierten Systemen inklusive ihrer Schnittstellen (Signalstruktur). MSCs (Message Sequence Charts, ISO Standard) ermöglichen die Definition von dynamischen Abläufen (vergleichbar Use Cases, Pfeildiagrammen) und damit die Basis von Testabläufen. Der Einsatz solcher Techniken integriert sehr gut die Möglichkeiten der Kode- und der Testgenerierung bzw. kontinuierliches Überprüfen von Systemmodellen.

Eine weitere Möglichkeit von TTCN ist die Spezifikation von verteilt auszuführenden Tests auf der Basis von parallelen Testkomponenten (z.B. je PCO) und ihre Synchronisation über eine Steuerkomponente und Koordinationsnachrichten. Hier sehen wir auch Möglichkeiten Schnittstellen in den automatisierten Test zu integrieren, die nicht Nachrichten- bzw. Signal-orientiert arbeiten.

Weitere Information:

[1] Martin Anlauf: *Programming service tests with TTCN*; Proceedings of the IFIP TC6 11th International Workshop on Testing of Communication Systems; Hrsg. Petrenko und Yevtushenko; Kluwer Academic Publishers, Boston, USA; 1998

[2] Jörn Münzel: *Improving the System Test Process*; Proceedings of the European Software Process Improvement Conference 1998 (EuroSPI ’98);