

Eine TTCN-3 Testplattform für reaktive Systeme

Ina Schieferdecker

Fraunhofer FOKUS/Technische Universität Berlin
Kaiserin-Augusta-Allee 31, D-10589 Berlin
schieferdecker@fokus.fraunhofer.de

Kurzfassung

Die Bereitstellung effizienter Methoden und Werkzeuge für die Bewertung und Kontrolle von zuverlässigen und vertrauenswürdigen Softwaresystemen ist ein Schlüssel zur weiteren Evolution der Softwaretechnologie. Heutige Softwaresysteme bestehen oftmals aus Komponenten, die in einer Net zumgebung verteilt sind und die Parallelitäts-, Autonomie-, Synchronisations-, Konfigurations- und Kommunikationsaspekte berücksichtigen müssen. Diese Systeme werden mit einer Vielzahl von Technologien (wie Middleware-Plattformen, Web Services, Messaging, etc.) realisiert, doch müssen sie letztendlich den Verfügbarkeits-, Integritäts-, Vertraulichkeits-, Zuverlässigkeit- und Verlässlichkeit anforderungen der Anwender gerecht werden. In TT-Medal wird eine Testplattform als zentrale Komponente für korrekte und zuverlässige Softwaresysteme erarbeitet: mit dieser Testplattform werden spezifikationsbasierte, formale Testmethoden gepaart mit automatischer Testgenerierung und -ausführung in die Systementwicklung integriert. Grundlage ist die formale Testmethode TTCN-3 (Testing and Test Control Notation). Die Testplattform wird auf ausgewählte Systeme der Automobil-Steuerung (wie Cockpit-Anwendungen basierend auf MOST/CAN Bussen) und der Mobilkommunikation (wie CORBA-basierte Netzwerksteuerung im UMTS) angewandt. In diesem Beitrag werden TTCN-3 und die Architektur der Testplattform mit ihren wesentlichen Komponenten beschrieben.

1. Einleitung

TT-Medal beschäftigt sich mit der Korrektheit, Sicherheit und Zuverlässigkeit von Softwaresystemen, indem der Softwareentwurfsprozess mit Methoden und Werkzeugen zum frühzeitigen, kontinuierlichen und systematischen Testen erweitert wird. Mit TTCN-3 wird das Testen von Systemanforderungen, Systemmodellen, Systemprototypen, als auch Systemimplementierungen unter Nutzung derselben Testspezifikationen bei modifizierter Testadaption ermöglicht. TTCN-3 wurde nicht nur für das Testen gesamter Systeme, sondern auch für das Testen von Modulen, Systemkomponenten und Teilsystemen konzipiert. In diesem Projekt wird eine Testmethodik für Black-Box und Grey-Box Tests auf Unit-, Integrations- und Systemebene basierend auf TTCN-3 entwickelt werden, die fortlaufendes, aufeinander aufbauendes Testen während der Systementwicklung unterstützt. Das schließt Ansätze zur Testgenerierung, Testtransformation und Testanalyse ein. Eine mit dem Entwurfsprozess verzahnte Testmethodik unterstützt das rechtzeitige Erkennen von Entwurfs-, Architektur- und funktionalen Fehlern, die dadurch kosteneffizient frühzeitig behoben werden können. Dies ist insbesondere für sicherheitskritische Software und eingebettete Systeme der Mobilkommunikation und Fahrzeugtechnik wichtig.

Im September 2000 wurde TTCN-3 durch ETSI - dem Europäischen Telekommunikationsstandardisierungs-institut - erstmals der Öffentlichkeit vorgestellt. Seither hat sich TTCN-3 sowohl im Anwendungs- als auch im Werkzeugbereich etabliert. Eine Vielzahl von Nutzern setzen TTCN-3 für die Entwicklung von Testlösungen für funktionale, Konformitäts-, Interoperabilitäts-, Skalierungs-, Lasttests, etc. für Protokolle, Dienste, objekt- und komponentenbasierte Systeme, für SmartCards, im Automobilbereich, etc. ein. Eine Palette von Werkzeugen verschiedener Anbieter steht dem Nutzer zur Verfügung. Eine Reihe von standardisierten Test Suites in TTCN-3 wie für SIP, SCTP und IPv6 oder aber Test Frameworks wie für CORBA basierte Systeme und Web Services können unmittelbar genutzt werden. Aber auch der Standard hat sich weiterentwickelt. Nach der Vorstellung der ersten Version von TTCN-3 (im Wesentlichen bestehend aus der Core Language und dem tabellarischen Präsentationsformat) sind als weitere Bestandteile die operationale Semantik, das graphische Präsentationsformat (GFT), die Ausführungsschnittstellen zur Anbindung an das zu testende System (TRI) und zur Anbindung an die Testplattform (TCI) als auch Einbettungen externer Typsysteme hinzugekommen. Damit steht mittlerweile ein stabiler und umfassender Ansatz zur abstrakten Testspezifikation- und -implementierung zur Verfügung. Ein kontinuierlicher Wartungsprozess für TTCN-3 bei ETSI ermöglicht die Korrektur, Verbesserung und den weiteren Ausbau von TTCN-3 entsprechend der Nutzererfordernisse.

TT-Medal wird in Deutschland im Rahmen der Software Engineering Offensive 2006 durchgeführt. Die deutschen Partner sind DaimlerChrysler (verantwortlich für die Anwendung der Testplattform in der Fahrzeugtechnik), Nokia (verantwortlich für die Anwendung der Testplattform in der Mobilkommunikation), Testing Technologies (verantwortlich für die Bereitstellung von TTCN-3 Werkzeugen) und Fraunhofer FOKUS (zuständig für die Erarbeitung der Testmethoden und Bereitstellung weiterer Testplattform-Komponenten). Darüber hinaus ist TT-Medal ein ITEA Projekt mit Partnern in Finnland und den Niederlanden. Einfluss auf die Arbeiten der deutschen Partner werden die finnischen Testgenerierungswerzeuge und die in Finnland und den Niederlanden durchgeführten Fallstudien haben.

2. Überblick zu TTCN-3

TTCN-3 ist die modernisierte Version der Testtechnologie TTCN zur Spezifikation und Implementierung von Tests. Sie unterstützt jegliche Art von Black-Box- und Grey-Box-Tests für reaktive, zentralisierte oder verteilte Systeme aus Telekommunikation, Mobilkommunikation, Informationstechnologie, eingebetteten Systemen, etc. Dabei unterstützt TTCN-3 verschiedene Arten von Tests wie funktionale Tests, Leistungstests, Skalierungs- tests, Interoperabilitätstests oder Konformitätstests.

TTCN-3 (siehe Abb. 1) besteht aus einer textuellen Kernsprache mit einer Schnittstelle zu verschiedenen Datenbeschreibungssprachen wie z.B. dem für Kommunikationsprotokolle genutzten ASN.1 (Abstract Syntax Notation One). Gleichzeitig erlaubt TTCN-3 die Anbindung verschiedener, auch grafischer Präsentationsformate an die Kernsprache. So sind im gegenwärtigen TTCN-3-Standard zwei Präsentationsformate definiert: ein tabellennotiertes Format in Anlehnung an die TTCN-3-Vorgängerversionen, und ein grafisches Format, das Testverhalten in Form von Sequenzdiagrammen visualisiert. Durch diesen Ansatz ist TTCN-3 unabhängig von konkreten Anwendungen und damit universell einsetzbar.

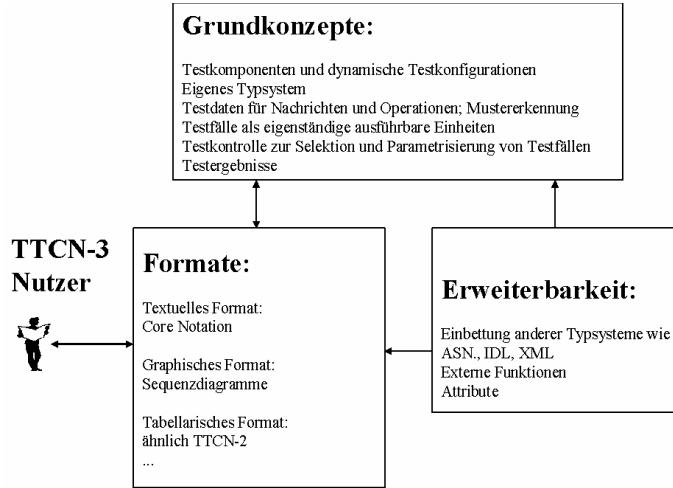


Abb. 1: TTCN-3 Überblick

In TTCN-3 existieren Konzepte, die die abstrakte Definition von Test Suiten, die unabhängig von konkreten Implementierungen des zu testenden Systems und des Testsystems sind, unterstützen und damit die Fähigkeiten von Programmiersprachen wie Java oder C erweitern. Werkzeuge wie Compiler und Ausführungsumgebung unterstützen die automatische Überführung der abstrakten Tests in ausführbare, die dann auf das zu testende System angewendet werden und die Testergebnisse liefern. Nach der Analyse der Testergebnisse erfolgt die Bewertung des getesteten Systems.

2.1. Grundkonzepte von TTCN-3

Beim Testen der Systeme wird das sogenannte Black-Box-Testen genutzt, wobei das zu testende System als Black-Box betrachtet wird und nur über seine Schnittstellen ansprechbar ist (analog zur Philosophie von Komponenten). Die Korrektheit des zu testenden Systems wird auf der Grundlage von Stimuli an das System und den daraufhin beobachteten Reaktionen an den Systemschnittstellen bewertet. Auf der Grundlage der beobachteten Reaktionen wird das weitere Vorgehen beim Testen bestimmt (z.B. welcher Stimulus als nächster zu senden ist) oder aber das Testergebnis zugewiesen. Dabei kann entweder ein Pass (bei diesem Testfall funktioniert das System korrekt), Fail (es konnte Fehler des Systems erkannt werden) bzw. Inconclusive (das Verhalten des Systems kann bzgl. des Testzwecks weder als korrekt noch als inkorrekt eingestuft werden). Zum Beispiel interessiert bei einem Web Server, ob eine Web Seite korrekt geladen werden kann, während es unerheblich ist, welche internen Abläufe für dieses Laden der Seite durchgeführt werden.

Während des Testens wird eine Reihe von Testfällen ausgeführt. Testfälle sind Testzwecken zugeordnet, die beschreiben, welche Systemeigenschaften mit Hilfe der Tests nachzuweisen sind. Diese Testzwecke werden auf der Grundlage der Systembeschreibung bestimmt. Testfälle definieren detailliert die Prozeduren, die notwendig sind, um den Nachweis der Systemeigenschaften zu führen.

Die Testfälle werden durch Testkomponenten ausgeführt. Es gibt eine ausgewählte Testkomponente – die Master-Testkomponente – die erzeugt wird, wenn immer ein Testfall gestartet wird. Diese Master-Testkomponente führt das Testverhalten aus, das für den Testfall definiert wurde. Jede Testkomponente kann weitere parallele Testkomponenten kreieren.

ren, so dass für verteilte Tests beliebig viele Testkomponenten genutzt werden können, um das Systemverhalten zu bewerten. Testkomponenten können über verbundene Schnittstellen (den so genannten Ports) Informationen austauschen bzw. mit dem zu testenden System interagieren. Die Menge der Testkomponenten und ihre Verbindungen untereinander und zum testenden System werden als Testkonfigurationen bezeichnet. Die Testkonfigurationen sind in TTCN-3 dynamisch, da während der Testausführung Testkomponenten kreiert und terminiert als auch Verbindungen auf- und abgebaut werden können. Dynamische Testkonfigurationen sind für die heutigen zu testenden Systeme wichtig, da auch diese im Allgemeinen während ihrer Laufzeit die Konfiguration dynamisch ändern, so dass das Testsystem im Verlaufe eines Tests angepasst werden muss.

Jede Testkomponente bewertet das zu testende System auf der Grundlage ihres lokalen Wissens. Ein Testfall terminiert, wenn die Master-Testkomponente beendet wird. Im Zuge dessen werden auch alle anderen Testkomponenten terminiert und die individuellen Testergebnisse zum Gesamtergebnis zusammengefasst. Dabei gilt die „es kann nur schlechter werden“-Klausel: wenn nur eine der Testkomponente Fail sagt, so ist das Gesamtergebnis Fail. Diese Regel sichert, dass Testergebnisse nicht „untergehen“ und nicht verfälscht werden können.

Die Definition von Testfällen erfolgt in TTCN-3 in sogenannten Modulen, wobei über ein vollständiges Modulkonzept der flexible Import von Definitionen anderer Module erlaubt ist. Bestandteile eines Moduls sind Deklarationen von Konstanten, Typen, Daten, Funktionen, Testverhalten, Testfällen und die Kontrollfunktion. Module können parametrisiert werden, um sie an spezielle Kontexte anpassen zu können.

An dieser Stelle sei für Details der angesprochenen Konzepte und weiterer Konzepte, die Syntax der TTCN-3 Kernsprache und Präsentationsformate auf den Standard verwiesen, welcher folgende Teile umfasst:

- Teil 1: TTCN-3 Kernsprache
- Teil 2: TTCN-3 Tabellarisches Präsentationsformat (TFT)
- Teil 3: TTCN-3 Graphisches Präsentationsformat (GFT)
- Teil 4: TTCN-3 Operationale Semantik
- Teil 5: TTCN-3 Runtime Interfaces (TRI)
- Teil 6: TTCN-3 Control Interfaces (TCI)

Zudem können in weiterführenden Papieren detaillierter Einführungen [2][4][8] und Beispiele [3][6][9] nachgelesen werden. Aktuelle Informationen zum Standard und der TTCN-3 Technologie sind unter www.etsi.org/ptcc/ptccttcn3.htm und www.testingtech.de/technology verfügbar. Der Nutzerkreis von TTCN-3 diskutiert über die eMail-Liste TTCN3@LIST.ETSI.ORG und trifft sich bei der TTCN-3 User Conference, wobei die erste dieser Konferenzen im Mai 2004 (siehe www.ttcn-3.org) stattfand. Die zweite TTCN-3 User Conference ist für Juni 2005 bei ETSI, Sophia-Antipolis, Frankreich geplant.

2.2. Die Ausführungsschnittstellen

Die Anpassung einer abstrakten TTCN-3 Test Suite an eine konkrete Testplattform und an ein zu testendes System zwecks Testdurchführung erfolgt über zwei standardisierten Schnittstellen TRI (TTCN-3 Runtime Interface) und TCI (TTCN-3 Control Interface). Ziel dieser Schnittstellenbeschreibungen (siehe Abb. 2) ist die Gewährleistung einer einfachen und wiederverwendbaren Anpassung einer TTCN-3-Test Suite an gegebene Test-Infrastrukturen.

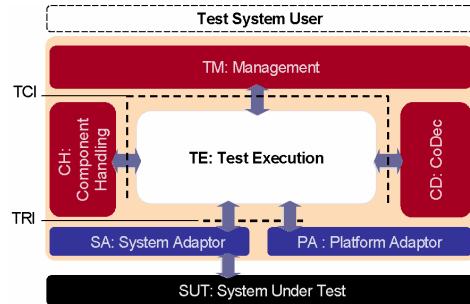


Abb. 2: TTCN-3 Ausführungsschnittstellen

Dabei definiert TRI im Wesentlichen die Anpassung an das zu testende System über

- den System Adaptor (SA) für die Kommunikationsschnittstellen und Kommunikation zwischen Testsystem und zu testendem System und
- den Platform Adaptor (PA) zur Implementierung von Timer-Operationen und externen Funktionen.

Dagegen realisiert TCI Schnittstellen zur Anpassung an das Testsystem/die Testplattform und bietet

- das Test Management (TM) zur Interaktion zwischen Nutzer und Testsystem während der Ausführung von TTCN-3 Tests
- das Component Handling (CH) zur Verteilung und Steuerung von Testkomponenten und zur Kommunikation zwischen Testkomponenten
- das Coding (CD) zum Enkodieren und Dekodieren von Testdaten.

Typischerweise werden die Realisierung von TCI-TM, TCI-CH und TRI-PA aus einer TTCN-3 Werkzeugumgebung wiederverwendet, während TCI-CD und TRI-SA entsprechend der Test Suite und dem zu testenden System angepasst werden. Dabei kann aber oftmals von existierenden TCI-CD und TRI-SA Adaptoren abgeleitet werden.

2.3. Der Import von externen Daten

TTCN-3 erlaubt den Import von externen Daten, die beispielsweise die Schnittstellen und Datenformate des zu testenden Systems definieren. Dieser Import kann sowohl explizit über einen vorgeschalteten separaten Transformationsschritt von den Daten nach TTCN-3 oder aber implizit über einen Import innerhalb eines TTCN-3 Moduls erfolgen. Dabei sind Abbildungen für

- Die Abstract Syntax Notation One (ASN.1) – eine in der Telekommunikation verwendete Datendefinitionssprache für Protokolle und Dienste und

- Die Interface Definition Language (IDL) – eine für objekt- und komponentenbasierte Softwaresysteme genutzte Schnittstellendefinitionssprache

Definiert worden. Eine Abbildung für Schemata der Extended Markup Language (XML) ist in Erarbeitung. Weitere Abbildungen können je nach Bedarf eines Nutzers hinzugefügt werden.

2.4. TTCN-3 in TT-Medal

Die in TT-Medal zu entwickelnde, TTCN-3 basierte Testplattform nutzt im Wesentlichen die Kernsprache, das graphische Format, die Semantik, die Ausführungsschnittstellen und den Import von ASN.1, IDL und XML Daten. Als Werkzeuge werden die von Testing Technologies eingesetzt. Zudem wird der TTCN-3 Standard entsprechend der Ergebnisse des Projekts weiterentwickelt. Hierzu zählen insbesondere die Beiträge zur Revision der Ausführungsschnittstellen, zum Import von XML Daten und zur Definition einer zusätzlichen Logging Schnittstelle.

Des Weiteren wird das bei der OMG entwickelte UML 2.0 Testing Profile (U2TP) mit seiner Abbildung nach TTCN-3 Anwendung finden. Dabei wird das Testgenerierungswerkzeug von Conformiq zur Ableitung von TTCN-3 Tests aus UML Modellen genutzt. Diese Integration von U2TP mit TTCN-3 ist eine weiterer Schwerpunkt standard-orientierter Arbeiten im Projekt.

3. Die TTCN-3-basierte Testplattform

In der Gesamtarchitektur der Testplattform wird zwischen Infrastruktur- und Plattformkomponenten unterschieden (siehe Abb. 3).

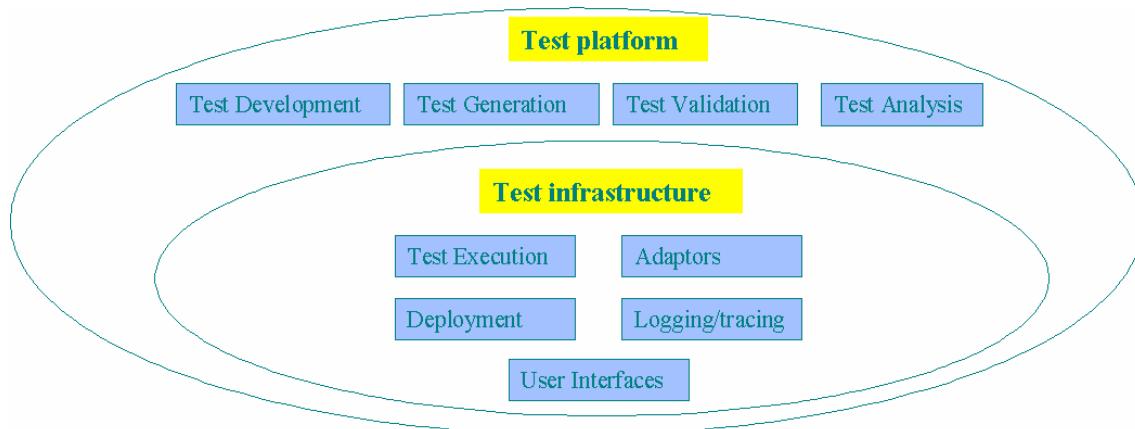


Abb. 3. Gesamtarchitektur der Testplattform

Die Testinfrastruktur umfasst alle Komponenten, die für die Ausführung von TTCN-3 Tests wesentlich sind. Diese Komponenten entsprechen den Komponenten eines TTCN-3 Testsystems, wie sie in TRI (Teil 5 von [1]) und TCI (Teil 6 von [1]) definiert sind (siehe auch 2.2). Darüber hinaus werden Komponenten für die Verteilung, Installation und Konfiguration von Tests auf einer verteilten Zieltestplattform und zum Aufzeichnen von Testausführungen vorgesehen. Die Steuerung der Testausführung und die Analyse der Ergebnisse soll-

te sowohl durch graphische Oberflächen (GUI) als auch durch kommando-orientierte Oberflächen (CUI) möglich sein.

Die Testplattform umfasst die Testinfrastruktur und weitere Komponenten zur Entwicklung und Wartung von Tests. Diese zusätzlichen Komponenten ermöglichen den Entwurf, die Generierung, Validierung und Analyse von Tests.

3.1. Testinfrastruktur-Komponenten

Die Testinfrastruktur (siehe Abb. 4) dient der Ausführung von TTCN-3 Tests und der Analyse der Testergebnisse. Sie wird zum Aufsetzen, zur Durchführung und Ergebnisdarstellung genutzt.

Abstrakte Test Suiten, die als Ergebnis der Testentwicklung entstehen, werden in ausführbare Tests mittels eines TTCN-3 Compilers (in unserem Fall TTthree) übersetzt.

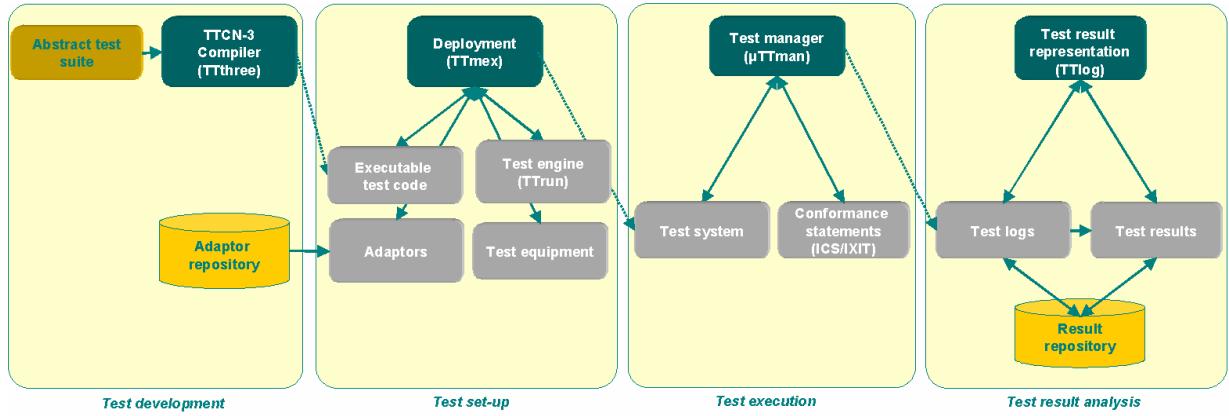


Abb. 4. Komponenten der Test Infrastruktur

Während des Aufsetzens von Tests (mittels TTmex) werden alle zur Ausführung notwendigen Test-Komponenten (lokal oder entfernt) verteilt und installiert. Diese Test-Komponenten enthalten den ausführbaren Code, die Testausführungskomponente (in unserem Fall TTrun), Adaptoren (zur Verknüpfung von Testsystem und zu testendem System, die oft aus Adaptor-Repositories genommen werden). Sie werden auf den Testgeräten (Spezial-Hardware oder aber normale Rechner, die über Demons als Testrechner für TTCN-3 ausgezeichnet werden) platziert.

Im Ergebnis des Deployments erhält man ein voll funktionsfähiges Testsystem, das mit dem zu testenden System verbunden ist, und die Testausführung ermöglicht. Das Testsystem wird entsprechend der Antworten im Implementation Conformance Statement (ICS) und Implementation Extra Information for Testing (IXIT) konfiguriert. Das Testmanagementwerkzeug μTTman wird zur Auswahl und Steuerung der Tests genutzt.

Im Ergebnis der Testausführung entstehen Test Logs und die finalen Testergebnisse. Diese können mit einer Testergebnisdarstellung (TTlog) visualisiert werden. Ebenso können Test Logs und die Testergebnisse in einem Results-Repository gespeichert werden.

3.2. Testplattform-Komponenten

Die Testplattform (siehe Abb. 5) bietet zusätzliche Komponenten zur Entwicklung und Wartung von Tests. Tests werden entsprechend der Testart (wie funktionales, Interoperabi-

litäts-, Leistungs-, Skalierungs-, Last-, Stress-, etc. Tests) und der für sie definierten Testziele erarbeitet. Tests können in TTCN-3, in domainen-spezifischen Profilen von TTCN-3 (wie für mobile Kommunikation oder für Fahrzeugtechnik), oder im UML 2.0 Testing Profile (welches dann auf TTCN-3 abgebildet wird) spezifiziert werden. Ergebnis der Testentwicklung sind abstrakte in TTCN-3 definierte Test Suiten, die in ausführbare Tests übersetzt werden. Diese werden dann mittels der Testinfrastruktur-Komponenten zur Ausführung gebracht.

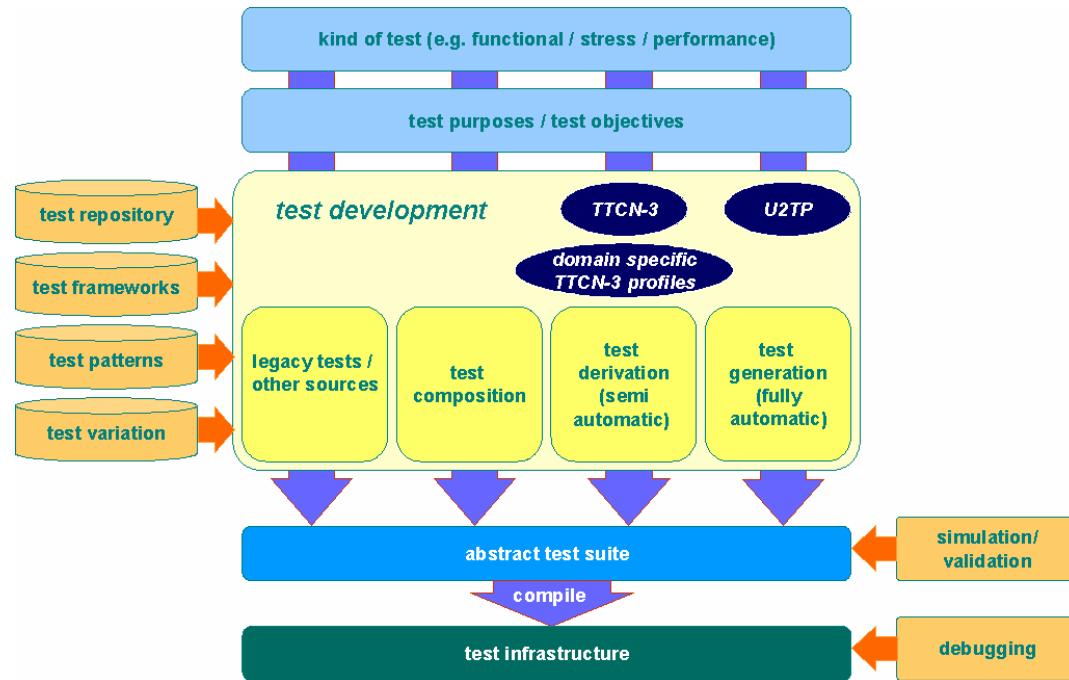


Abb. 5. Komponenten der Test-Plattform

Tests können unter Nutzung existierender Tests (aus hausinternen, anderen Systementwicklungsprozessen oder aber von anderen Quellen wie beispielsweise Standardisierungsgremien) aus diesen abgeleitet werden oder manuell entwickelt werden. Zudem können Tests aus anderen Tests komponiert werden oder aber aus Anforderungsspezifikationen, System-szenarien oder Systemmodellen semi-automatisch oder voll-automatisch abgeleitet werden.

Existierende Tests stehen typischerweise über Test-Repositories zur Verfügung. Bei der Testkomposition werden Test Patterns und Test Frameworks genutzt. Test Pattern definieren sich wiederholende Muster in der Testkonfiguration, im Testverhalten oder in den Testdaten. Test Frameworks bieten spezifische, für eine Domäne, Technologie oder aber eine konkrete Systemfamilie/ein konkretes System angepasste Testdefinitionen an. Test Pattern können durch Instanzierung konkretisiert werden. Test Frameworks werden durch Parametrisierung angepasst. Verfeinerung kann sowohl für Test Patterns als auch für Test Frameworks genutzt werden.

Des weiteren können abstrakte Test Suiten unter Nutzung der Testplattform-Komponenten auf Korrektheit und Konsistenz geprüft werden. Dazu können Komponenten zur Simulation, Validierung und zum Debugging eingesetzt werden.

5. Ausblick

Das TT-Medal Projekt erarbeitet derzeit die prinzipiellen Testabläufe und Testmethoden. Dabei werden Ansätze für Test Patterns, Test Frameworks und domänen-spezifischen Testprofile, deren Anwendung, Verfeinerung und Validierung erarbeitet. Diese Ansätze werden Einfluss auf die Definition der Testplattform-Komponenten und deren Entwicklung haben. Testgenerierungsansätze wie die Kombination von UML mit TTCN-3 (über das UML 2.0 Testing Profile und das Conformiq Werkzeug) und die Kombination der Klassifikationsbaummethode mit TTCN-3 (über das DaimlerChrysler Werkzeug CTE) erarbeitet. Diese Ansätze werden zudem die Testentwicklung effektivieremn.

Gleichzeitig werden bereits die Fallstudien in der Mobilkommunikation und der Fahrzeugtechnik entwickelt. Nach einer ersten Demonstration von Testinfrastruktur-Komponenten beim ITEA Symposium im Okt. 2003 wird beim nächsten ITEA Symposium im Okt. 2004 die Nutzung der Testplattform zur Analyse von MOST basierten Cockpit-Anwendungen demonstriert werden.

Literatur

- [1] ETSI: The Testing and Test Control Notation TTCN-3, Part 1-6: ETSI ES 201873-1 bis ETSI ES 201873-6, V2.2.2, Oct. 2003.
- [2] J. Grabowski, D. Hogrefe, G. Rethy, I. Schieferdecker, A. Wiles, C. Willcock: An Introduction into the Testing and Test Control Notation (TTCN-3). - Computer Networks Journal, Vol.42, Issue 3, 2003
- [3] I. Schieferdecker, S. Pietsch, T. Vassiliou-Gioles: Systematic Testing of Internet Protocols - First Experiences in Using TTCN-3 for SIP. 5th IFIP Africom Conference on Communication Systems, Cape Town, South Africa, May 2001.
- [4] I. Schieferdecker, T. Vassiliou-Gioles: Realizing distributed TTCN-3 test systems with TCI, IFIP 15th Intern. Conf. on Testing Communicating Systems - TestCom 2003, Cannes, France, May 2003.
- [5] T. Vassiliou-Gioles, G. Din, I. Schieferdecker: Execution of External Applications using TTCN-3. IFIP 16th Intern. Conf. on Testing Communicating Systems - TestCom 2004, St Anne's College, Oxford, United Kingdom, March 2004.
- [6] I. Schieferdecker: System-Level Tests with TTCN-3. - Proc. Of World Conference on Integrated Design and Process Technology, IDPT, Austin, Texas, Dec. 2003.
- [7] I. Schieferdecker, G. Din: A Metamodel for TTCN-3, Akzeptiert für 1st International Workshop on Integration of Testing Methodologies (ITM 2004), Oct. 2004, Toledo, Spain, colocated Workshop of 24th IFIP Intern. Conf. On Formal Techniques for Networked and Distributed Systems FORTE 2004, Sept. 2004, Madrid, Spain.
- [8] I. Schieferdecker, T. Vassiliou-Gioles: Sicher wie ein Telefon - TTCN-3: Eine neue Testmethode für IT-Systeme. - iX 11/01, Seite 56 ff, 2001.
- [9] I. Schieferdecker, B. Stepien: Automated Testing of XML/SOAP based Web Services, 13. Fachkonferenz der Gesellschaft für Informatik (GI) Fachgruppe "Kommunikation in verteilten Systemen" (KiVS), Leipzig, 26.-28. Febr. 2003.