TEST SYSTEM ARCHITECTURES USING ADVANCED STANDARDIZED TEST LANGUAGES

Axel Rennoch
InSTA 2016, Chicago, 10th April 2016

Fraunhofer FOKUS
CONTENTS

- Introduction

- Advanced standardized test languages
  - TTCN-3
  - UML testing profile
  - TDL

- Test Automation Architecture

- Conclusions
AXEL RENNOCH

- Researcher at Fraunhofer FOKUS in Berlin, Germany

- User of standardized Modelling and Test languages
  - LOTOS, SDL/UML
  - Tree and Tabular Combined notation (TTCN-1&2)

- TTCN-3 user from the beginning
  - Developer for ETSI, 3GPP and industry
  - Trainer and consultant
  - Research projects and evolution team
MOTIVATION

InSTA 2015:
  – Keynote by Sigrid Eldh on
    – Software Test Architecture
    – definition & aspects

InSTA 2016:
  – Continuation on Advanced Standardized Test Languages for
    – Test System Architecture
    – means & examples, user perspective
CHALLENGES FOR TEST SYSTEMS

Data type integration
Execution mapping
Logical structure & distribution
SUT interfaces (physical distribution)

maschine languages  programming languages  test description languages  test modelling languages
SOME INITIAL QUESTIONS FOR TEST DEVELOPERS

- Identify SUT access interface points
- Test objectives: functional, load, security?

- Target:
  - standardization body (more abstraction, top down) or
  - in-house test solution (address concrete tools, bottom up)

- Parallel test components (scalability?)
- Coordination & synchronization (existing frameworks?)
CONTENTS

Introduction

Advanced standardized test languages
  - TTCN-3
  - UML testing profile
  - TDL

Test Automation Architecture

Conclusions
WHAT IS TTCN-3?

- Testing and Test Control Notation
- Internationally standardized testing language for formally defining test scenarios. Designed purely for testing

```plaintext
testcase Hello_Bob () {
 p.send("How do you do?");
 alt {
     []p.receive("Fine!");
     {setverdict( pass )};
     [else]
     {setverdict( inconc )} //Bob asleep!
 }
}
```
IDEA & DESIGN PRINCIPLES OF TTCN-3

- One test technology for different tests
  - Distributed, platform-independent testing
  - Integrated graphical test development, documentation and analysis
  - Adaptable, open test environment

- Areas of Testing
  - Regression Testing
  - Conformance and Functionality Testing
  - Interoperability and Integration Testing
  - Load/ Stress Testing
testcase Hello_Bob () {
    p.send("How do you do?");
    alt {
        []p.receive("Fine!");
        {setverdict( pass )};
        [else]
        {setverdict( inconc )} //Bob asleep!
    }
}
TTCN-3 IS DESIGNED FOR BLACK-BOX TESTING

TTCN-3 Test Case

Port.send(Stimulus)  Port.receive(Response)

• Assignment of a Test Verdict

System Under Test
A TTCN-3 TEST SYSTEM

Test System User

- TM: Management
- TL: Logging
- CD: Codec
- CH: Component Handling
- SA: System Adapter
- PA: Platform Adapter

System Under Test (SUT)

ETSIs:
- ETSI ES 201 873-1: TTCN-3 Core Language (CL)
- ETSI ES 201 873-5: TTCN-3 Runtime Interface (TRI)
- ETSI ES 201 873-6: TTCN-3 Control Interfaces (TCI)

Abbreviations:
- TE – TTCN-3 Executable
- TM – Test Management
- TL – Test Logging
- CD – Codec
- CH – Component Handling
- SA – System Adapter
- PA – Platform Adapter
- SUT – System Under Test
MAJOR LANGUAGE ELEMENTS OF TTCN-3 NOTATION

module definitions

<table>
<thead>
<tr>
<th>Imports</th>
<th>Importing definitions from other modules defined in TTCN-3 or other languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Types</td>
<td>User defined data types (messages, PDUs, information elements, …)</td>
</tr>
<tr>
<td>Test Data</td>
<td>Test data transmitted/expected during test execution (templates, values)</td>
</tr>
<tr>
<td>Test Configuration</td>
<td>Definition of the test components and communication ports</td>
</tr>
<tr>
<td>Test Behavior</td>
<td>Specification of the dynamic test behavior</td>
</tr>
</tbody>
</table>
TTCN-3 DOMAINS: TELECOM

- Industrial use
  - Big companies with hundreds of TTCN-3 engineers: Ericsson, Nokia, Siemens, Motorola
  - Large distribution among SME
- Standardization bodies
  - Standardized test suites:
    - ETSI / 3GPP (LTE) / OMA / TETRA and its members
  - IMS performance benchmark project:
    - Intel, HP, BT, FOKUS and others
- Test tool manufacturer:
  - Commercial Tektronix, Catapult, Nexus, R&S, Spirent, ...
- Certification program based on TTCN-3: e.g. WiMax forum
Secure UserPlane Location Protocol

**Single MTC controls e.g.:**

- UlpPort (Lup interface)
- IpcPort (IP configuration)
- SmsPort used for SMS
- UtpPort for upper tester commands
- IpiPort (IP information, e.g. release)
- NwcPort: network bearer control, e.g. handover trigger
- SscPort: satellite simulation control, e.g. scenario trigger
TEST SYSTEM EXAMPLE 2: 3GPP UE TESTING

- **E-UTRAN (LTE air interface):** 3GPP TS 36.523-3 V12.4.0 (2015-12)

- Each radio access technology (RAT) is hosted by a separate TTCN-3 parallel component: E-UTRAN, UTRAN, GERAN, others.

- PTCs are controlled by the TTCN-3 main test component (MTC) which:
  - is independent from the RAT;
  - may host the upper tester for MMI and AT commands;
- Multiple of configurations
- Several PTCs
  - Cooperating
  - Communicating
TEST SYSTEM EXAMPLE 2: 3GPP UE TESTING

- Simple scenario
- Reuse of nested component types for MTC, PTCs and TSI
- More complex configuration
- Nested component types
- Illustration using Visual Paradigm
TTCN-3 DOMAINS: AUTOMOTIVE

- Car communication systems
  - Daimler, Volkswagen, SiemensVDO
  - edutainment bus system (test suite)
- Standardization groups:
  - AUTOSAR consortium
  - MOST cooperation
- Car-to-car communication

Telematics Applications in the Cockpit

- Audio (CD / Radio), Video
- Telephone, SMS
- Navigation
- Speech recognition
- User interface for body electronic

Test cases: loudness playCDTitle

Fraunhofer FOKUS
TTCN-3 DOMAINS: MEDICINE

Medicine
- SiemensMED (image processing)
- HL7 eHealth protocols (Interoperability)

Upcoming E-Health infrastructure for Germany
- High security requirements (e.g. certificates, cryptography)
- Test development prior to SUT availability
- Multiple heterogenous interfaces:
  - cardterminals, card simulations,
  - Webservices, OCSP server etc.
TEST SYSTEM EXAMPLE 3: E-HEALTH “KONNEKTOR”

- Complex configuration

- Illustration using IBM Rational Enterprise Architect:
  - PTC,
  - simulators,
  - SUT (mock),
  - real server,
  - adaptation code
- **Source** STF 276 (IPv6 project)

- Set of **TTCN-3 functions** to e.g.
  - Start and control parallel components
  - Exchange synchronization signals between components

- Set of **charstring constants** for **synchronization points**
  - `preambleDone`, `sync1`, `sync2`
  - `testbodyDone`

- **Predefined timers** e.g. to avoid deadlocks at synchronization points

Synchronization occurs between parallel TTCN-3 components (using signals via MTC)

- Sync 1 ensures the completion of senders procedure
- Sync 2 confirms the arrival of a message at receivers side
MODULE IMPORT „LAYERING“

- Test suite specific code
  - testcase, test (component) functions

- Test suite specific library
  - Component types, test configuration, pre/postamble

- Interface/protocol specific library
  - SIP, DNS, IPv6 etc.

- Interface/protocol independent library
  - synchronizaton

high level:
  (low reusability)

Low level:
  (High reusability)
EXAMPLE 4: IMS BENCHMARK TEST CONFIGURATION

 Sender Component

 Handler 1 Component

 Handler 2 Component

 Load Generator

 System Adapter (SUT Interface)

 SUT

- 400 subscribers per component **Handler**
- Component **Sender** initializes/provides data requests for load-generator
- 5.000 - 10.000 IMS subscribers (per server)
- Up to 250 requests per second (per server)
LESSONS LEARNED WITH TTCN-3

- Study access interface points (Test System Interface)
  - Available Test solution plugins?
- Test objectives: functional, load, security?
  - Consider synchronization overhead (e.g. ETSI’s LibCommon)
  - Consider performance issues (e.g. encoded data preparation)
- Concurrent test components with separated traces & verdicts (easier failure analysis)
  - Consider test tool logging functions
- Improve decomposition using libraries for
  - handling single interface types
  - Nested component types (to be extended)
- Use modelling tools for the illustration of the TTCN-3 architecture
BEYOND TTCN-3: MBT

- TTCN-3 is used in several domains as binding link between modelling and execution

- Commercial tools do generate TTCN-3 code for test execution
  - lots of academic prototype tools

- Selection of industrial case studies: e.g. European MIDAS project
  - pilots in SOA testing automation (later)
**Objective:**
- To develop an efficient **test platform** fulfilling **industrial testing requirements**
- To **execute high-level test models**, e.g. UML testing profile

---

**TTCN-3 LINK TO UTP**

- Industrial Testing Requirements
  - Embedded Systems
  - Automotive
  - Mobile Communication
  - Avionics
  - Railway Systems

---

**Approach**
- UML 2.0 Testing Profile
- TTCN-3

---

**An efficient test platform**
- Methods, Tools, Guidelines

---

**System Design**
- Test Design
- Test Results
CONTENTS

- Introduction

- Advanced standardized test languages
  - TTCN-3
  - UML testing profile
  - TDL

- Test Automation Architecture

- Conclusions
UML TESTING PROFILE (UTP)

- **Standard by the Object Management Group:**
  Version 1.0 (2004), current version 1.2 (2013), revised *draft* version 2.0 (Nov 2015)

- **Profile of UML version 2:**
  Industrial standard for *(graphical)* modeling of Test architectures, behavior and data.

- **Conceptual Model:**
  Test context, cases, objectives, data, configuration, arbitration&verdicts, logs

- **UTP library:**
  predefined types and values (e.g. ISO 25010 Quality model, ISTQB Test levels)

- **Extras:**
  Mapping to TTCN-3 *(procedure-based communication only)*
TEST CONFIGURATION OVERVIEW

- Standardized mapping of UTP stereotypes to UML metaclasses

```plaintext
(<<metaclass>>) StructuredClassifier
    «extends»
    «extends»

(<<metaclass>>) Classifier
    «extends»
    «extends»

(<<metaclass>>) Property
    «extends»
    «extends»

(<<metaclass>>) Constraint
    «extends»
    «extends»

(<<stereotype>>) {concepts = test configuration}
    Test Configuration
    ID : String

(<<stereotype>>) {concepts = test item}
    Test Item

(<<stereotype>>) {concepts = test component}
    Test Component

(<<stereotype>>) {concepts = test item configuration}
    Test Item Configuration

(<<stereotype>>) {concepts = test component configuration}
    Test Component Configuration
```
UTP TEST CONFIGURATION EXAMPLE (UTP 2.0 ONLY!)

- LoginServer
test **components**

- LoginServer
test **configuration**
UTP MAJOR USE

- Domain-independent **test modeling for dynamic testing** approaches:
  - Test environments, test configurations, test case specifications (including test case derivation), test data specifications/values

- Test **evaluation**, i.e., managing and visualization of **test results**

- **Integration of best practices**
  such as keyword-driven testing, equivalence class testing, etc.

- **Combination with other UML profiles** (e.g., SysML, MARTE, SoaML)
  - E.g. to achieve requirements traceability, …
LESSONS LEARNED WITH UTP

- **Current version 1.2**
  - less industrial use since version 1.0 **ten years ago**
  - No big test suites
  - Only some tooling

- **UTP Revision 2.0 is promising**
  - E.g. covers test configuration
Introduction

Advanced standardized test languages
- TTCN-3
- UML testing profile
- TDL

Test Automation Architecture

Conclusions
TEST DESCRIPTION LANGUAGE

- New ETSI Standard
  ES 203 119-1 (V1.2.0, 2015-04)

- Fills gap between the high-level test purposes and TTCN-3

- Simple Text notation with graphical presentation
Typed components and gates
Timers and variables
connections among gates
component roles

Gate Type gt accepts Login, Response;
Component Type ct having {
gate g of type gt;
}
Test Configuration tc {
create Tester tester of type ct;
create SUT sut of type ct;
connect tester.g to sut.g;
}
LESSONS LEARNED WITH TDL

- Pure testing view
- Compromise between UTP and TTCN-3
  - Simple
  - Executable
- Not ready to use
Introduction

Advanced standardized test languages
- TTCN-3
- UML testing profile
- TDL

Test Automation Architecture

Conclusions
THE ISTQB FUNDAMENTAL TEST PROCESS

Management
Test Analysis
Test Design
Test Implementation
Test Execution
Test Evaluation
Test Closing Activities

Automated test design (MBT)

Knowledge
Test model
Test generator
Test generator

Model transformation

Test Scripts
Test execution tool
System under test

Automated test execution (KDT, DDT, TTCN-3)

Automated test execution is state of the practice (if ever) in industry

Fraunhofer FOKUS
ABSTRACTION LEVELS IN TEST AUTOMATION

Test design activities
- Technical test cases (ABS. LEVEL 2)
  - Test model (ABS. LEVEL 3)
- Logical test cases (ABS. LEVEL 3)

Test analysis activities
- SRS (ABS. LEVEL 4)

Test implementation activities
- Executable test cases
  - Technical test cases (ABS. LEVEL 2)
  - Adaptation layer (ABS. LEVEL 2->0)
- Actual implementation (ABS. LEVEL 0)

Test execution activities

Logical actions the system can perform

Sequence diagram, Test Descriptions
Implementation, TRI, SA, CoDEC

UML Testing Profile, TDL
KDT, DDT, TTCN-3
TTCN-3, XML, Excel

Fraunhofer FOKUS
TOP-DOWN APPROACH

Model is master
- Test design independent from adaptation layer or test execution system
- No constraint on the test execution system
- Often used in academic prototypes

Recommendation
Feasible for proof-of-concepts, limited use for industry
Adaptation layer is master
- Ensures immediate automated test execution
- **Requires** available adaption layer
- Test model derived from the adaption layer

**Recommendation**
*Only if* adaptation layer is clear for test developers
TEST AUTOMATION ARCHITECTURE: MIDAS

SOA System

WSDL/XSD

import

exhibits

MIDAS Test Model

Functional

Fuzzing

Usage-based

MIDAS TPaaS

Test Generation Layer

MIDAS Test Model + Test Cases

MDTA Framework

Test Definition Layer

TTCN-3 Modules (complete)

Test Execution Layer

Fraunhofer FOKUS
Two approaches have been shown

- Bottom-up approach was realized in MIDAS

- **Integration of data types** (e.g. WSDL) is challenging

- (Initial) engineering effort can be quite high
CONTENTS

- Introduction

- Advanced standardized test languages
  - TTCN-3
  - UML testing profile
  - TDL

- Test Automation Architecture

- Conclusions
<table>
<thead>
<tr>
<th></th>
<th><strong>TTCN-3</strong></th>
<th><strong>UTP</strong></th>
<th><strong>TDL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standardization body</strong></td>
<td>ETSI, ITU-T</td>
<td>OMG</td>
<td>ETSI</td>
</tr>
<tr>
<td><strong>History</strong></td>
<td>Since 1992</td>
<td>Since 2004</td>
<td>Since 2014</td>
</tr>
<tr>
<td><strong>Applicability</strong></td>
<td>All domains and testing types</td>
<td>All domains</td>
<td>Conformance, interop</td>
</tr>
<tr>
<td><strong>Execution tools and solutions</strong></td>
<td>+</td>
<td>proprietary via C-unit</td>
<td>TTCN-3 mapping in preparation</td>
</tr>
<tr>
<td><strong>Current user groups</strong></td>
<td>Industry, Research</td>
<td>Academic, Research</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>TTCN-3</td>
<td>UTP 2</td>
<td>TDL</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
<td>--------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Component extension</td>
<td>+</td>
<td>UML generalization</td>
<td>- (reuse elements)</td>
</tr>
<tr>
<td>Coordination/synchronization</td>
<td>+ via libraries</td>
<td>using general ordering</td>
<td>explicitly</td>
</tr>
<tr>
<td>Import of WSDL, IDL, etc.</td>
<td>(+)</td>
<td>proprietary via SoaML</td>
<td>-</td>
</tr>
<tr>
<td>Graphical Test architecture</td>
<td>- needed!</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Link to UML</td>
<td>-</td>
<td>+</td>
<td>in preparation</td>
</tr>
</tbody>
</table>
GOOD REASONS FOR STANDARDIZED TEST LANGUAGES

– They significantly increase your **system quality**.
– You can focus on what to test, not on how.
– They reduce costs and efforts in test system **maintenance**.
– They are **independent** of access technology, operating system and implementation domain.

– They support **communication** between system development and test department.
– You can count on available, trained and **certified experts**.
CONCLUSION

- **TTCN-3, UTP and TDL**
  - international Standards for testing
  - allow abstract definitions for testing
  - (partly) accepted in research and industry

- **Tool support** (still week)
  - UML -> UTP/TDL -> TTCN-3

- Test automation need further enhancements
TTCN-3 SOURCES

- Online information → www.ttcn-3.org
- TTCN-3 User Conference → 2016 in Budapest, Hungary
- TTCN-3 Standards, Papers, Book → http://www.ttcn.de/
- Quick Reference → http://www.blukaktus.com/
- Exercises and Tooling → research licenses
- Online information → http://utp.omg.org/
- MBT User Conference → 2016 in Budapest, Hungary
- MDT/UTP Book → http://www.springer.com
- Exercises and Tooling → research licenses

TDL User Conference ➔ 2016 in Budapest, Hungary


Exercises and Tooling *(in preparation)* ➔ [https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx](https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx)
Thank you for your attention!

www.fokus.fraunhofer.de
(System Quality Center)
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
www.fokus.fraunhofer.de

Axel Rennoch  
Project Manager  
axel.rennoch@fokus.fraunhofer.de  
Phone +49 30 3463-7344

Marc-Florian Wendland  
Senior Researcher  
marc-florian.wendland@fokus.fraunhofer.de  
Phone +49 30 3463-7395