INTRODUCTION TO TTCN-3

Presented by Theofanis Vassiliou-Gioles
Theofanis Vassiliou-Gioles
Founder and CEO of Testing Technologies
vassiliou@testingtech.com
www.testingtech.com

- Master in Electrical Engineering
- Started communication testing 1996
- ATM test specification standardization
- ETSI TTCN-3 standardization
- Application of test automation in new domains
Agenda

- Motivation
- History
- TTCN-3 Standards
- TTCN-3 by Example
- Module Definitions
- Test Case Execution
- Summary
INTRODUCTION TO TTCN-3

Motivation
How Much Does Testing Cost?

"... the national annual cost estimates of an inadequate infrastructure for software testing are estimated to be $59.5 billion. The potential cost reduction from feasible infrastructure improvements is $22.2 billion."

The Economic Impacts of Inadequate Infrastructure for Software Testing

Study by NIST, May 2002
And Today?

World Quality Report 2013-14

As consumers demand high performance, error-free applications, organizations are increasing their QA budgets and more testing functions are centralized.

Focus on Testing is growing everywhere...
A higher share of the IT budget is invested in Testing...

- 23% of the IT budget is spent on QA & Testing - compared to 18% last year -

46% of QA budget is spent on "transformational projects" - compared to 41% last year -

...organizations are industrializing and outsourcing their QA...

...and as mobile applications increase, mobile testing gains traction...

- 41% of projects are done entirely in-house - compared to 51% last year -

- 26% of organizations have a centralized testing function - up from 8% in 2012 -

- 19% of businesses have fully operational Test Centers of Excellence - compared to 6% last year -

PRIMARY FOCUS
**PRIMARY FOCUS**

- #1 Efficiency and Performance: 59%
- #2 Security: 56% - up from 18% last year -

**BIGGEST CHALLENGE**

Lack of appropriate processes/methods

34% → 56%

...and cloud-based testing is expected to increase.

By 2015, 32% of Testing will be performed in the Cloud

30% of cloud-based testing is performed on critical, externally facing applications - up from 20% last year -
Spendings in Testing (WQR 2013)

- Hardware: 40%
- Software: 28%
- Rest / People: 32%
What is TTCN-3?

- **Testing and Test Control Notation**
- Internationally standardized testing language for formally defining test scenarios. Designed purely for testing
- In its essence it can be considered as a kind of scripting language that includes tons of testing specific features!

```plaintext
testcase tc_Hello_Bob () {
  p.send("How do you do?");
  alt {
    [] p.receive("Fine!"){
      setverdict( pass );}
    [else]{
      setverdict( inconc );}  //Bob asleep!
  }
}
```
testcase tc_Hello_Bob () {
   p.send("How do you do?");
   alt {
      [] p.receive("Fine!"){
         setverdict( pass ); }
      [else]{
         setverdict( inconc ); }
   }
}
Why Using TTCN-3 (1)
Why Using TTCN-3 (2)
Why Using TTCN-3 (3)

- Speed to Market
- High Integration
- High Reliability
- Optimal Cost
- Complex (WiFi, LTE)
- High Quality
- MATCH
- Requirements
- Test Methods
INTRODUCTION TO TTCN-3

History
History (1)

- **TTCN (1992)**
  - Published as an ISO standard
  - Tree and Tabular Combined Notation
  - Used for protocol testing only
    - GSM, N-ISDN, B-ISDN

- **TTCN-2/2++ (1997)**
  - Concurrent tests
  - Modularization
  - Manipulate external data
  - Rather for conformance testing
  - Developed by ETSI MTS
History (2)

  - Testing and Test Control Notation
  - Developed by ETSI MTS
  - Standard language
    - Well defined syntax and semantics
  - Enhanced communication, configuration and control
  - Standard test specification
    - SIP, SCTP, M3UA, IPv6
    - HiperLan, HiperAccess, Wimax
    - 3GPP LTE, OMA
    - TETRA
    - MOST, AUTOSAR
    - EUROCONTROL
### History (3)

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>Start of developing TTCN-3</td>
</tr>
<tr>
<td>2000</td>
<td>TTCN-3 Standard /TT Foundation</td>
</tr>
<tr>
<td>2002</td>
<td>ETSI: SIP</td>
</tr>
<tr>
<td></td>
<td>ETSI: HiperACCESS</td>
</tr>
<tr>
<td></td>
<td>ETSI: HiperLAN2</td>
</tr>
<tr>
<td>2003</td>
<td>ETSI: HiperMAN</td>
</tr>
<tr>
<td>2004</td>
<td>WiMAX Forum: 802.1d</td>
</tr>
<tr>
<td></td>
<td>ETSI IPv6</td>
</tr>
<tr>
<td>2005</td>
<td>WiMAX Forum: 802.1e</td>
</tr>
<tr>
<td></td>
<td>AUTOSAR: Pilot</td>
</tr>
<tr>
<td></td>
<td>WiMAX Forum: Interop</td>
</tr>
<tr>
<td></td>
<td>ETSI: VoIP/POTS Testing</td>
</tr>
<tr>
<td>2007</td>
<td>OMA: Interop Pilot</td>
</tr>
<tr>
<td></td>
<td>AUTOSAR: Rollout</td>
</tr>
<tr>
<td>2008</td>
<td>OMA: Rollout</td>
</tr>
<tr>
<td>2011</td>
<td>TETRA: Pilot / Rollout</td>
</tr>
<tr>
<td>2012</td>
<td>EUROCONTROL: Rollout</td>
</tr>
</tbody>
</table>

- Since 2002 standard bodies using TTCN-3 to define test specifications
  - ETSI 3GPP
  - WiMAX Forum
  - OMA
  - TETRA
  - AUTOSAR
  - MOST
  - EUROCONTROL
Testing is...

- a technical process
- performed by experimenting with a software product
- in a controlled environment
- following a specified procedure
- with the intent of observing one or more characteristics of the product
- by demonstrating the deviation of the product’s actual status from the required status/specification.
Testing Today’s Systems

- Component-based
  - Test-components contribute to SUT functionality and performance
- Distributed
  - Not only local, but also distributed test setups
- Dynamic in terms of behavior and configuration
  - Testing of static and dynamic aspects; dynamic creation of test components
- Use various type systems to exchange data
  - Open to all type systems
- Service is essential
  - Concentration on service-oriented black-box testing
Design Principles of TTCN-3

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

• One test technology for distributed IT and telco systems and beyond
INTRODUCTION TO TTCN-3

TTCN-3 Standards
TTCN-3 Standards

- ETSI ES 201 873-1  TTCN-3 Core Language (CL)
- ETSI ES 201 873-2  TTCN-3 Tabular Presentation Format (TFT)
- ETSI ES 201 873-3  TTCN-3 Graphical Presentation Format (GFT)
- ETSI ES 201 873-4  TTCN-3 Operational Semantics
- ETSI ES 201 873-5  TTCN-3 Runtime Interface (TRI)
- ETSI ES 201 873-6  TTCN-3 Control Interfaces (TCI)
- ETSI ES 201 873-7  Integration of ASN.1
- ETSI ES 201 873-8  Integration of IDL
- ETSI ES 201 873-9  Integration of XML
- ETSI ES 201 873-10 T3Doc
- ETSI ES 202 781  TTCN-3 Extension: Configuration And Deployment Supp
- ETSI ES 202 782  TTCN-3 Extension: Performance & Real-Time Testing
- ETSI ES 202 784  TTCN-3 Extension: Advanced Parametrization
- ETSI ES 202 785  TTCN-3 Extension: Behaviour Types
- ETSI ES 202 786  TTCN-3 Extension: Continuous Signals
- ETSI ES 202 789  TTCN-3 Extension: Extended TRI

- Maintenance on the basis of change requests by ETSI
- Standard available for download at http://www.etsi.org
- Testing Tech tools support Edition 4.7.1
- Also standardized by the ITU-T as ITU-T Z.16x series
INTRODUCTION TO TTCN-3

TTCN-3 by Example
TTCN-3 By Example

Tester

Local Network Client

Main Test Component

Send fully qualified hostname

Return IP-address

Local Domain Name Server

System Under Test

User Conference on Advanced Automated Testing
TTCN-3 By Example

Tester

Client

pass

Main Test Component

System Under Test

DNS

(www.testingtech.com,A)

(www.testingtech.com,87.106.91.172,A)
## Generic Protocol Architecture(s)

<table>
<thead>
<tr>
<th>OSI View</th>
<th>TCP/IP View</th>
</tr>
</thead>
<tbody>
<tr>
<td>L7 / Application</td>
<td>HTTP, FTP, SMTP, POP, Telnet, DNS</td>
</tr>
<tr>
<td>L6 / Presentation</td>
<td>TCP, UDP, SCTP</td>
</tr>
<tr>
<td>L5 / Session</td>
<td>IPv4, IPv6</td>
</tr>
<tr>
<td>L4 / Transport</td>
<td>Ethernet, Token Bus, Token Ring, FDDI, IPoAC</td>
</tr>
<tr>
<td>L3 / Network</td>
<td></td>
</tr>
<tr>
<td>L2 / Data Link</td>
<td></td>
</tr>
<tr>
<td>L1 / Phy</td>
<td></td>
</tr>
</tbody>
</table>
Generic Protocol Architecture(s)

OSI View

TCP/IP View

20-22/10/2015

User Conference on Advanced Automated Testing
INTRODUCTION TO TTCN-3

Module Definitions
TTCN-3 Modules

- Main building block of TTCN-3 is a module
  - Unit of compilation
  - Contains definitions
  - Optional control part

```c
module DNS {

  // module definitions

  // module control (optional)

}
```
Module Definitions

• Contains descriptions for
  • What type of data the System Under Test understands
  • How the System Under Tests can be accessed and what environment a test component needs
  • When to communicate what with the SUT and why
  • Dependencies between test cases, if any
Module Definitions (1)

- Module definitions
  - Type definitions
  - Port definitions
  - Component definitions
  - Templates
  - Test case
- Control part
  - Controls the execution of test cases

```plaintext
type record DNSQuery {
    charstring hostname,
    AnswerType answer optional,
    QueryType qtype
}

type union AnswerType {
    Byte ipAddress[4],
    charstring hostname
}

type integer Byte (0 .. 255);

type enumerated QueryType {
    A, NS, CNAME, MX
}
```
Module Definitions (2)

- Module definitions
  - Type definitions
  - Port definitions
  - Component definitions
  - Templates
  - Test case
- Control part
  - Controls the execution of test cases

Port definitions

```plaintext
type port DNSPort message {
  inout DNSQuery;
  // a port may send/receive messages
  // of more than one type
}
```

Component definitions

```plaintext
type component DNSTester {
  port DNSPort P;
  timer t := 3.0;
  // a component may have more than one port
}
```
Module Definitions (3)

- Module definitions
  - Type definitions
  - Port definitions
  - Component definitions
- Templates
- Test case

- Control part
  - Controls the execution of test cases

```plaintext
template DNSQuery query := {
    hostname := "www.testingtech.com",
    answer   := omit,
    qtype    := A
}
template DNSQuery reply modifies query := {
    answer := { ipAddress :=
        {87,06,91,172} }
}
```

"www.testingtech.com"  A

"www.testingtech.com"  87, 06, 91, 172  A
Module Definitions (4)

- Module definitions
  - Type definitions
  - Port definitions
  - Component definitions
  - Templates
  - Test case

- Control part
  - Controls the execution of test cases

```
testcase tc testcase1() runs on DNSTester {
  P.send(query);
  P.receive(reply);
  setverdict(pass);
}

// there may be more than one in a module
```

User Conference on Advanced Automated Testing
Module Definitions (5)

- Module definitions
  - Type definitions
  - Port definitions
  - Component definitions
  - Templates
  - Test case

- Control part
  - Controls the execution of test cases

```plaintext
control {
    execute(tc_testcase1(), 5.0);
    while( /* condition */ ) { }

    // more testcases might follow
    // C-like control structures available
}
```
INTRODUCTION TO TTCN-3

Test Case Execution
Execution of a Test Case

```c
// Tester

MTC
verdict = pass

P

query = (www.testingtech.com,A)

P

reply = (www.testingtech.com, 87.106.91.172,A)

SUT

Tester

testcase tc_testcase1() runs on DNSTester {
    P.send(query);
    P.receive(reply);
    setverdict(pass);
}
```

Is this test case definition adequate?
Is this an effective test case definition?
Dealing with Erroneous Behavior (1)

• **P.receive** (*reply*) blocks until it receives a message that matches the reply

• If unexpected message is received, any other correct message does not unblock the tester, which then blocks forever

• If no message is received, the tester will also block forever
Dealing with Erroneous Behavior (2)

```java
testcase tc_testcase2() runs on DNSTester {
    P.send(query);
    t.start;
    alt {
        [] P.receive(reply) {
            setverdict(pass);
        }
        [] P.receive { // any message
            setverdict(fail);
        }
        [] t.timeout {
            setverdict(inconc);
        }
    }
    stop;
}
```

Is it an effective test case definition now?
Code Reusability – Altsteps and Defaults

```
alt {
    [] P.receive(reply) {
        setverdict(pass);
    }
    [] P.receive { // any message
        setverdict(fail);
    }
    [] t.timeout {
        setverdict(inconc);
    }
}
```

becomes

```
var default d := activate(a_RefactoredAltstep());
P.send(query);
t.start;
P.receive(reply);
setverdict(pass);
```

refactor

```
altstep a_RefactoredAltstep() {
    runs on DNSTester {
        [] P.receive { // any message
            setverdict(fail);
        }
        [] t.timeout {
            setverdict(inconc);
        }
    }
}
```
Non-Local DNS Query

Tester

Send fully qualified hostname

Return IP address

Remote DNS (NS)

Parallel Test Component 3

Ask for remote DNS

Get remote address

Parallel Test Component 2

The internet’s root name service (root NS)

Local Network Client (Client)

Parallel Test Component 1

Main Test Component

System Under Test

Local Domain Name Server (DNS)

20-22/10/2015 User Conference on Advanced Automated Testing
From Simple To Complex Test Scenarios

- Test system needs more interfaces
  - Test System Interface has to be extended
- Test behavior required at additional test interfaces
  - Behavior of Local Network Client already covered in `tc_testcase2`
  - Behavior of RootNS and NS required
- Test case that combines all parts
Parallel Test Components

- Test system interface

```plaintext
type component TestSystemInterface {
    port DNSPort CLIENT;
    port DNSPort ROOT;
    port DNSPort NS;
}
```
From Test Case to Behavior Function

- Functions can be used to define the behavior of the parallel test components

```cpp
testcase tc_testcase2() runs on DNSTester {
    var default d := activate(a_refactoredAltstep());
    P.send(query);
    t.start;
    P.receive(answer);
    setverdict(pass);
    stop;
}
```

becomes

```cpp
function f_clientBehavior() runs on DNSTester {
    var default d := activate(a_refactoredAltstep());
    P.send(query);
    t.start;
    P.receive(answer);
    setverdict(pass);
    stop;
}
```
Additional Test Behavior

• Simple „react-on-request“ behavior

```javascript
function f_rootBehavior() runs on DNSTester {
    alt {
        [] P.receive(rootquery) {
            P.send(rootanswer);
            setverdict(pass);
        }
        [] P.receive {
            setverdict(fail);
        }
    }
}

function f_nSBehavior() runs on DNSTester {
    alt {
        [] P.receive(nsquery) {
            P.send(nsanswer);
            setverdict(pass);
        }
        [] P.receive {
            setverdict(fail);
        }
    }
}
```
Dynamic Configuration

testcase testcase3() runs on DNSTester
system TestSystemInterface {

var DNSTester ClientComp, RootComp,
    NSComp;

ClientComp := DNSTester.create;
RootComp := DNSTester.create;
NSComp := DNSTester.create;

map(ClientComp:P, system:CLIENT);
map(RootComp:P, system:ROOT);
map(NSComp:P, system:NS);

ClientComp.start(f_clientBehavior());
RootComp.start(f_rootBehavior());
NSComp.start(f_nSBehavior());

ClientComp.done;
    // block until ClientComp is done
    stop;
}

User Conference
on Advanced Automated Testing
Some Geek Details

• Case sensitive
  • More than 130 (edition 4) keywords, all lower case
  • Identifiers

• Comments
  • Multi line comments: /* */
  • Single line comments: //

• Statements are terminated with: ;

• Statement blocks are enclosed in: {   }

• Operators
  • Assignment: :=
  • Comparison: !=, ==, <=, >=
INTRODUCTION TO TTCN-3

Summary
TTCN-3 Summary
Why Should I Use TTCN-3?

• To have
  • an industrial grade
  • multi-os and
  • multi-technology
• middleware testing platform
  • to build your methods
  • to create supporting tools and tool chains
  • to be able to educate people
• in order to
  • cover the whole software development process
FAQs

- After learning about TTCN-3, what do I have to do to make it really work?
- Is TTCN-3 only for [conformance | protocol] testing?
- Can I use TTCN-3 to test embedded systems?
- Why should I use TTCN-3 instead of my [java, python, perl, ...] tool?
- How complicated is it to learn TTCN-3?
- Does anybody care if I know TTCN-3?
- Where can I find more information on TTCN-3?
- What is the relation between TTCN-3 and [insert here your favorite]
- What is the difference to my selenium test tool?
- What is the biggest TTCN-3 installation in a company?
- What is the smallest TTCN-3 installation in a company?
- I have to do GUI testing. Can you please show me how I could use TTCN-3 to accomplish this?
- And what have you learned?
Some References

• The language
  - www.ttcn-3.org
  - de.wikipedia.org/wiki/TTCN-3
  - en.wikipedia.org/wiki/TTCN-3
  - t-ort.etsi.org

• TTCN-3 Training
  - www.ttcn-3.org/index.php/learn/tutorials
  - testingtech.com/services/ttcn3_training.php

• The Quick Reference Card
  - www.blukaktus.com/card.html

• Some tools
  - www.ttcn-3.org/index.php/tools/tools-com