Advanced Test Modelling and Execution Based on the International Standardized Techniques TTCN-3 and UTP

Axel Rennoch[✉], Marc-Florian Wendland, Andreas Hoffmann, and Martin Schneider

Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
`{axel.rennoch,marc-florian.wendland,andreas.hoffmann, martin.schneider}@fokus.fraunhofer.de`

Abstract. In systems and service engineering testing is an important part to get confidence in quality and trust in security issues. Standardized testing techniques support the unique definition of abstract test models, configurations and behavior scenarios that can be executed automatically. This contribution presents the state of the art and future directions of two international standards for testing: the Testing and Test Control Notation (TTCN-3) from the European Telecommunication Standardization Institute (ETSI), and the UML testing profile (UTP) from the Open Management Group (OMG). Special emphasize is given to the translation from UTP to TTCN-3 test models, automated test execution using standard-compliant tool support and related examples from European projects.

Keywords: Test automation · Model-based test design · TTCN-3 · UML · UTP

1 Introduction

In the technical world of today testing is crucial for quality assurance, expensive (costs!) and time critical but only rarely practiced, often unsystematic and performed by hand, error-prone and even considered as uncool ("If you are a bad programmer you might be a tester.") or destructive.

The idea of advanced test modelling and execution is to have one abstract test technology for different kinds of tests, including distributed, platform-independent testing, integrated graphical test development, -documentation and analysis, and an adaptable, open test environment for test campaigns.

All areas of testing should be addressed, e.g. regression, conformance and functionality, interoperability, integration, security, load/stress testing and even monitoring and benchmarking shall be covered.

2   Testing and Test Control Notation Version 3 (TTCN-3)

2.1   Concepts

TTCN-3 is the standardized test language and implementation published and maintained by the European Telecommunication Standardization Institute (ETSI) for formally defining test scenarios [4]. It has been designed purely for testing. TTCN-3 is both a specification technique and an implementation language. Abstract test definitions can be compiled into executable test programs written e.g. in Java, C or C ++. The specification is intended to be understood also by non-programmers. There are even different presentation formats available, e.g. for the text-based or graphical-oriented core notations. Most developers use the text-based notation that may be translated by some tools (e.g. TTworkbench [12]) into the graphical notation in order to illustrate e.g. test behavior steps or functions into a kind of well-known message sequence chart. A simple test case in textual notion as given in the following figure can be an input to the TTCN-3 based tester to communicate with a system under test (SUT) (Fig. 1).
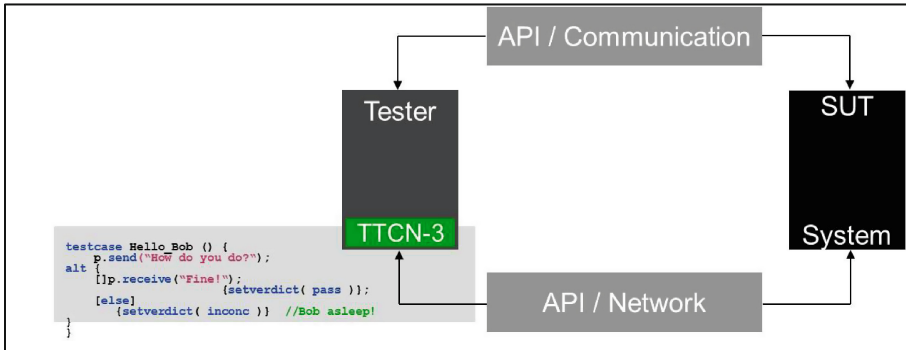

Fig. 1. Test execution using a TTCN-3 test system

In addition to the language notation, TTCN-3 technology provides much more: There is e.g. the standardized test system architecture with clearly defined interfaces and implementation rules for the primitives, but also various mapping rules for the integration of different data types or values specified with ASN.1, IDL or XML. Currently the standardized mapping from JSON to TTCN-3 is under development at ETSI and is planned to become the new part eleven of the TTCN-3 multipart standard in the 2015. A quick overview about the TTCN-3 types, programming constructs and predefined functions etc. is provided in the TTCN-3 Quick Reference Card that is freely available by BluKaktus [10].

2.2   TTCN-3 Major Use

TTCN-3 has a long history starting in 1992 with a first TTCN standard from ISO until it gains the great value and meaning of today, e.g. for the 3GPP mobile conformance

center test specifications used by the Global Certification Forum [1]. The TTCN-3 technology is not restricted to telecom conformance testing and has a wide range of applicability, different communication paradigms and testing types. Actually it is used for multiple industrial domains that include telecom, automotive, medicine, finance, railways, avionics, etc. in both research and industry.

3 UML Testing Profile (UTP)

3.1 Concepts

The UTP Standard by the Object Management Group started in 2004 with its version 1.0, is currently published in its version 1.2 [9] from 2013, and is on the road to new version 2.0 that has been drafted and proposed in May 2014. UTP is based on UML version 2, and provides an industrial standard for (graphical) modeling of e.g. test architectures, behavior and data.

Its conceptual model includes and allows objects for test context, test cases, objectives, data, configuration, arbitration & verdicts, and logs. UTP offers a library with predefined types and values (to support e.g. ISO 25010 System and software quality model [7] or ISTQB test levels) and a mapping to TTCN-3.

3.2 UTP Major Use

Similar to TTCN-3, UTP offers domain-independent test modeling for dynamic testing approaches including 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. It allows the integration of best practices such as keyword-driven testing, equivalence class testing, etc. and can be combined with other UML profiles (e.g., SysML, MARTE, SoaML) in order to support requirements traceability.

4 A Common View on TTCN-3 and UTP

It is not the intention of this paper to present or reflect the history or details of the various technical comparisons between TTCN-3 and UTP, since this exists for a long time and includes lots of excellent detailed work as given in [11] and [3] or [6]. In the industrial user perspective it appears that UTP has been developed after the establishment of TTCN-3 as kind of a derivate in some parts but UTP is on a much higher level of abstraction. UTP also tried to fill some gaps in TTCN-3 like the emphasis of the test environment and a generalization of some concepts that are present in TTCN-3 due to its historical root from earlier TTCN versions.

The integration of UTP and TTCN-3 focusses and allows the development of an efficient test platform fulfilling industrial testing requirements, ideally to execute high-level test models from UML testing profile.

As illustrated in Fig. 2, TTCN-3 is used in several domains as a binding link, i.e. a kind of middleware, between modelling and execution. Please note that there exist several tools to generate TTCN-3 code for test execution.
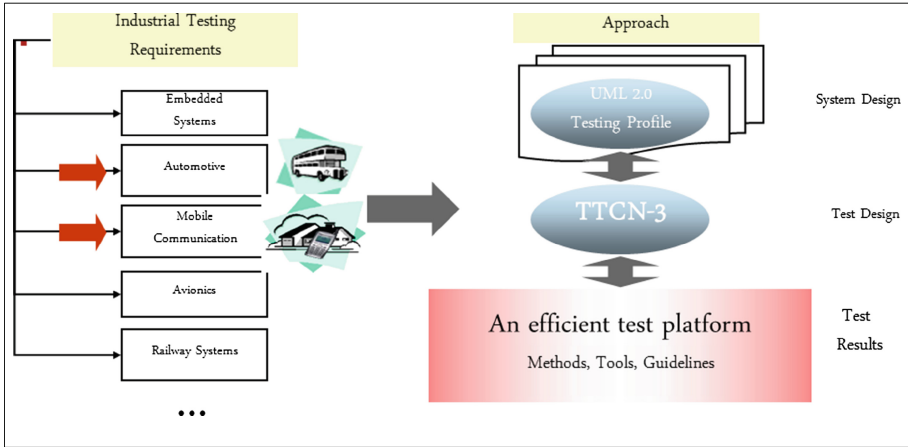

Fig. 2. Common approach using TTCN-3 and UTP

5   Tools and Applications

The acceptance of test techniques often depends on tools and experiences. Here we provide some well-known tools and applications, but also a short presentation of the European MIDAS project working for automated testing of Services architectures, available as a Software as a Service (SaaS) on a Cloud infrastructure. This project spans all activities including test generation, scheduling, execution, and arbitration and it relies on the standards TTCN-3 and UTP.

5.1   Industrial Perspective

First of all, the industrial application requires professional tool support and services. Today, there exist several high quality commercial tools on the market that include compilers/interpreters but also utilities with special focus on the e.g. generation of TTCN-3 code from high level model specifications. The following lists provide a selection retrieved from our knowledge and the TTCN-3 homepage at ETSI [4].

TTCN-3 Compilers and Interpreters

- Exhaustif/TTCN: compiler (C ++) produced by Métodos y Tecnología (MTP), Spain.
- OpenTTCN: interpreter (C, Java, C# interfaces) produced by OpenTTCN Ltd, Finland.
- Testcast: compiler (C/C ++, Java, C#) produced by ELVIOR, Estonia.

- Real Time Developer Studio: modelling tool including TTCN-3 compiler by PragmaDev, France.
- TAU Tester: compiler (C) by IBM.
- TTCN-3 Toolbox: compiler (C) by Danet Group, Germany.
- TTCN-3 Express: compiler (C#) by Fraunhofer FOKUS and Metarga GmbH, Germany.
- TTworkbench: compiler (C, Java) by Testing Technologies, Germany.

TTCN-3 Generators

- Qtronic by Conformiq OY, Finland, generate complete TTCN-3 test suites from e.g., UML, Java, or C# models.
- MaTeLo by All4Tec, France (TTCN-3 test suites from usage models specified using Markov chains).
- MOTES by ELVIOR, Estonia (from the state model of the SUT).

It is also possible to select one of the open source tools listed in the next section. However, customers need to carefully check the level of standard compliance, scope of the implementation and last but not least the professional vendor support from the tool provider. Under some circumstances (e.g. in standardization bodies) it is recommended to apply different tools in order to allow a wide distribution of the developed TTCN-3 scripts.

In the following, we summarize some short collection about the industrial usage of TTCN-3 technology:

Industrial use
Telecom industry

- Big telecom companies with hundreds of TTCN-3 engineers (e.g. Ericsson, Nokia, Siemens, Motorola) and a large distribution among SME.
- Standardization bodies: standardized test suites by ETSI, 3GPP (LTE!), OMA, TETRA and its members.
- IMS performance benchmark project by Intel, HP, BT, FOKUS, and others.
- Commercial test tool manufacturer like Spirent, Tektronix, Catapult, Nexus, R&S.
- Certification programs based on TTCN-3: e.g. WiMax forum.

Car communication systems

- Daimler, Volkswagen, SiemensVDO: edutainment bus system (test suite).
- Standardization groups like AUTOSAR and MOST cooperation, e.g. Car-to-car communication.

Other domains

- Medicine: SiemensMED (image processing), HL7 eHealth protocols (Interoperability).
- Power transmission and distribution: SiemensPTD (safe and reliable energy system).
- Financial data warehouse: International bank (functional /regression testing).
- Avionics: European Space Agency.
- Railways: Dutch railways (TT-Medal project [14]).

5.2   Research Perspective

From the researcher's viewpoint open source tools, free utilities and innovative research project results are of great value. The following provides a short selection of non-commercial tools listed due to our knowledge. Furthermore we provide a short introduction of an innovative framework for the integration of UTP and TTCN-3.

TTCN-3 Non-commercial tools

- LoongTesting testing platform including TTCN-3 compiler and integrated development environment by Information Processing Center of USTC, China.
- BBT TTCN-3 Compiler by BroadBit, Hungary.
- TRex by University of Göttingen that provides IDE functionality for TTCN-3 core notation, and support the assessment and automatic restructuring of TTCN-3 test suites (open-source Eclipse plug-in).
- T3doc by F. Engler and further developed by ETSI for generating HTML documentation via tagged TTCN-3 comments.
- Codec generator by IRISA as part of T3DevKit. It automatically generates a codec based on TTCN-3 type module(s), C ++ codec functions.
- T3DevLib by IRISA as part of T3DevKit. It allows the development or integration of Codec, SUT and Platform Adapter implementations written in C ++.
- Titan, a brand new Open source community project driven by Ericsson experts [13].

There are even more academic prototype/research tools, e.g. for guideline checking, quality analysis. Since there is not such big list of UTP research tools we like to put your attention to the Fokus!MBT framework [5] that allows some integration of the most important test development activities including automated test case synthesis and test execution (cp. Fig. 3).

5.3   The MIDAS Project

The European research project MIDAS [8] is developing a Cloud platform for automated testing of SOA services. For this purpose, model-based test generators, schedulers, and arbiters for functional and inference-driven testing, usage-based testing, and security testing are provided as services running on the MIDAS platform. The platform also provides dedicated services for test code generation and execution. To achieve this, all services rely on a UML profile that is based on UTP 1.2 with some advancement. Thus, all generator services generate models that are compliant to this MIDAS domain-specific language (DSL). The experiences with and the advancements of UTP 1.2 made in the MIDAS research project led to immediate contributions to the version 2 draft of UTP.

The test execution service of the MIDAS platform is completely based on TTCN-3 and corresponding compiler, plugins, and codecs are developed and gratefully provided by Testing Technologies IST GmbH. The model-based test cases compliant to the UTP-based MIDAS DSL are transformed to TTCN-3 and then executed against the service under test.
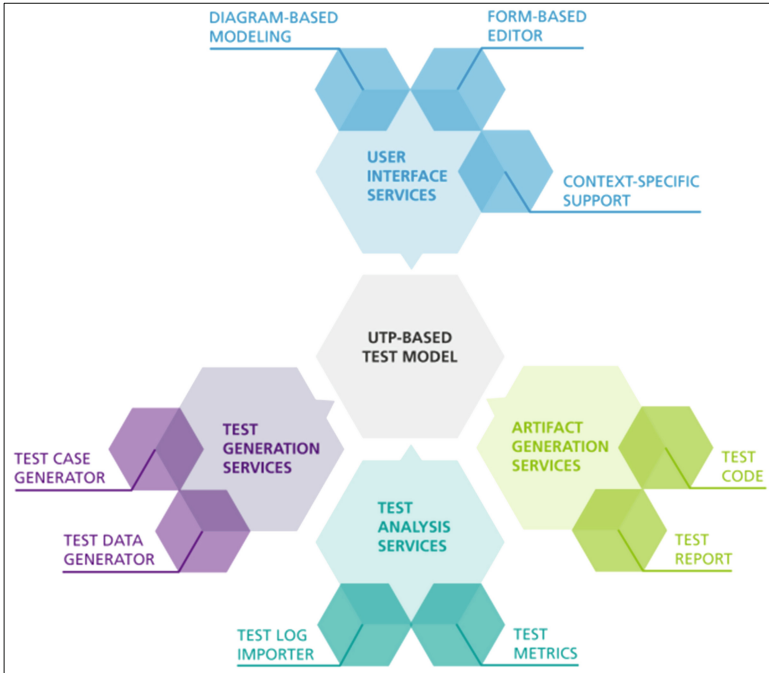
Fig. 3. Fokus!MBT

In 2015 the MIDAS project is in its third and last project year and thus, in its evaluation phase with the help of its pilots from the eHealth and the Logistics domain.

6   Outlook

TTCN-3 created a large international community with great expertise (see e.g. [15]), commercial and non-commercial tools. Today, TTCN-3 is established and well accepted for future usage. Most hand written scripts are easily, human readable description of test scenarios, free from programming issues, providing a transparent framework for end-customers that allows platform and tool vendor independence, with different presentation formats. The clear separation of testing issues allows writing of well-defined test configuration, data and behavior, as well as SUT specific adaptations and codecs.

UTP has been recognized by several communities but is still waiting for a larger field of application. The situation may change quickly in 2015 with the upcoming UTP version 2.

References

1. 3rd Generation Partnership Project (3GPP), RAN5 Mobile Competence Center, Task force 160: http://www.3gpp.org/about-3gpp/mobile-competence-centre/mcc-task-forces
2. Ali, S., et al.: How does the UML testing profile support risk-based testing. In: 2014 IEEE International Symposium on Software Reliability Engineering Workshops. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6983859
3. Dai, Z.-R.: An Approach to Model-Driven Testing – Functional and Real-Time Testing with UML 2.0, U2TP and TTCN-3, Ph.D. thesis, Fraunhofer FOKUS, Berlin (2006)
4. European Telecommunication Standards Institute (ETSI): ES 201 873: Testing and Test Control Notation, version 3 (TTCN-3). http://www.ttcn-3.org
5. Fraunhofer FOKUS: Fokus!MBT. http://www.fokusmbt.com
6. Herbold, S., et al: MIDAS project deliverable D3.6: Generating TTCN-3 from PIM UML SAUT Specifications (2014)
7. International Standards Organization: ISO/IEC 25010, Systems and software engineering - System and software quality models, Geneva (2011)
8. MIDAS project consortium. http://www.midas-project.eu
9. Object Management Group (OMG): UML Testing Profile. http://www.omg.org/spec/UTP
10. Rennoch, A., et al.: TTCN-3 Quick reference card. www.blukaktus.com
11. Schieferdecker, I., et al.: The UML 2.0 testing profile and its relation to TTCN-3. Testing of Communicating Systems. LNCS 2644, 2003, pp. 79–94. Springer, Berlin (2003)
12. Testing Technology: TTworkbench. http://testingtech.com/products/ttworkbench.php
13. Titan, Eclipse-based IDE for TTCN-3. https://projects.eclipse.org/projects/tools.titan
14. TT-Medal project consortium. https://itea3.org/project/tt-medal.html
15. TTCN-3 Bibliography: http://www.ttcn.de/bibliography