Int J Softw Tools Technol Transfer (2014) 16:215–225
DOI 10.1007/s10009-014-0302-9

INTRODUCTION

History, status, and recent trends of the testing and test control notation version 3 (TTCN-3)

With a brief introduction to selected articles from the TTCN-3 user conference 2011

Jens Grabowski · Ina Schieferdecker · Andreas Ulrich

Published online: 15 March 2014

Abstract This overview article presents the *Testing and Test Control Notation* (TTCN-3) success story and serves as an introduction to this Special Section that contains five articles selected from the TTCN-3 user conference in 2011. The article sketches the development of TTCN-3 from its very beginning. It summarizes the current status of the language by reviewing its standardization process, available test suites, tools, and services as well as its training program. In addition, the article puts the articles selected for this Special Section into perspective, with regard to the evolution of TTCN-3 and the testing methodology in general. Last but not least, it discusses indicators for possible future developments of TTCN-3.

Keywords TTCN-3 · Test specification · Test automation · Test architecture · Test execution · Standardization

1 Introduction

The *Testing and Test Control Notation* (TTCN-3) is a modern language for test specification and implementation by means of which tests can be designed, specified, and documented textually or graphically and can be implemented, executed, and logged automatically. TTCN-3 was developed by the *European Telecommunications Standards Institute* (ETSI) as a new edition of the *Tree and Tabular Combined Notation version 2* (TTCN-2). TTCN-3 was mainly designed for conformance, interoperability, and performance testing of communicating systems including communication protocols, Web services, as well as system and component interfaces of IT systems.

TTCN-3 is used throughout the testing life-cycle by test designers, testers, test managers, quality managers, and project managers. TTCN-3 has gained acceptance in the industry for testing the conformance and interoperability of business-, security- and safety-critical systems. Yet, the adoption of TTCN-3 as the basis for test automation and quality assurance has certain implications on the core test process. For example, explicit test specifications in TTCN-3 have to be established as central artifacts. TTCN-3 test specifications need to be designed, developed, maintained, and evolved along changing requirements.

In addition, since TTCN-3 tools provide powerful support for the automated execution of TTCN-3 tests and the analysis of the test results, testers get relieved from the tedious task of test execution and can spend more time on the addition of more advanced and elaborated tests in order to improve the overall quality of the *System Under Test* (SUT).

TTCN-3 has evolved from a mere test language to a comprehensive test technology used in various application areas such as telecommunications, avionics, and automotive. It has gained support by several commercial, open source, and academic tool sets and test suites.

This article presents the TTCN-3 success story and also introduces the articles selected from the TTCN-3 User Conference 2011 for this Special Section. The article recaps the language history in the next section, which is followed by a presentation of its current status in Sect. 3. Last but not least,

J. Grabowski (✉)
University of Goettingen, Goettingen, Germany
e-mail: grabowski@informatik.uni-goettingen.de;
grabowski@cs.uni-goettingen.de

I. Schieferdecker
FU Berlin/Fraunhofer FOKUS, Berlin, Germany
e-mail: ina.schieferdecker@fokus.fraunhofer.de

A. Ulrich
Corporate Technology, Siemens AG, Munich, Germany
e-mail: andreas.ulrich@siemens.com

Sect. 4 provides the authors' view on the future of TTCN-3 before the article is concluded in Sect. 5.

2 A journey to the roots of TTCN-3

2.1 The beginning

The history of TTCN-3[1] dates back around 30 years with an earlier version that was known in that time as the *Tree and Tabular Combined Notation* (TTCN). This version was closely related to the evolution of the *Open Systems Interconnection* (OSI) reference model [31] and the associated ISO/IEC standard 9646: *OSI Conformance Testing Methodology and Framework* (CTMF) [32]. The OSI reference model defines the communication protocols for implementing interconnected and communicating systems from different manufacturers on seven different layers of abstraction. Conformance testing was defined as a prerequisite to assure the interoperability of OSI protocol entities. CTMF provides the means for standardized conformance testing, including terminology, test methods, test procedures, and also TTCN [33].

In 1983, Dave Rayner from the National Physical Laboratory in the UK was appointed as rapporteur for the upcoming CTMF standard and the associated working group started to define the conformance testing concepts. The group recognized soon that an unambiguous notation for test specification was required. Consequently, the first requirements for a test notation were investigated and collected. These requirements included OSI-specific concepts, such as *Abstract Service Primitive* (ASP) and *Protocol Data Unit* (PDU), test-specific concepts like verdicts and test methods, as well as OSI-specific test requirements such as single-layer testing, embedded single-layer testing, and multi-layer testing.

A first mature draft of TTCN was ready by 1985 and, as often the case in standardization, it implemented a compromise. The TTCN tree syntax for behavior description was influenced by the *Formal Description Technique* (FDT) *Language of Temporal Ordering Specification* (LOTOS) [30] and tabular formats popular in several areas such as teletext.

As the transfer syntax for TTCN specifications, a TTCN BNF was developed by the Institute of Computer Science in Kista, Sweden. This syntax was subsequently standardized as the *TTCN Machine Processible Form* (TTCN.MP), whereas the tabular form was called *TTCN Graphical Form* (TTCN.GR). TTCN.MP acted not only as a transfer syntax, but also served as a formal TTCN syntax. TTCN.MP eased the automatic derivation of *Executable Test Suites* (ETS) from *Abstract Test Suites* (ATS) and enabled all other kinds of machine processing. In parallel to the development of TTCN.MP, the first TTCN tool, called *Interactive TTCN Editor and Executor* (ITEX), was developed.

By 1990, the TTCN draft became more stable, but also larger and larger. A major effort was put into implementing a close connection to the *Abstract Syntax Notation One* (ASN.1) [34], an important data description language in the telecommunications domain. In addition, an unambiguous operational semantics was developed.

2.2 The evolution of TTCN

It took two more years until, in 1992, TTCN became an official ISO/IEC standard. This TTCN version, also called TTCN-1, was published as part 3 of ISO/IEC IS 9646 [33] and in parallel as ITU-T recommendation X.292. TTCN-1 was widely adopted by the GSM community. Tool support was improving and the first TTCN tutorial was published [36].

Based on the first experiences with TTCN-1, the next TTCN version, called TTCN-2 was published in 1993. It added parallelism to TTCN and provided an improved operational semantics. TTCN-2 was accepted as conformance test specification language for most of the important telecommunications technologies, such as ISDN, GSM, DECT, and INAP. Formalized conformance testing became more and more popular, leading to the emergence of conformance testing services initiated by the European Commission and offered by third-party testing providers.

By 1995, testing expertise and TTCN had migrated to ETSI. The TTCN development and maintenance in ISO had ceased and was taken up by the *ETSI Technical Committee for Methods for Testing and Specification* (ETSI TC MTS). In 1999, ETSI published the last update of TTCN-2, called TTCN-2++ [24]. It sought to remove flaws and improved minor features based on practical experiences with TTCN-2 at ETSI.

2.3 The TTCN-3 idea

Although there had been significant legacy in TTCN-2, it was nonetheless recognized that the times were changing: on the one hand, the Internet-way of testing became popular, i.e., ad hoc interoperability testing, rather than formalized conformance testing and certification. On the other hand, mobile communications, complex radio technology, and new application areas required rigorous testing. TTCN was adequate, but limited in its use and application. In addition, programming trees in tables had no appeal to the new generation of testers. Hence in 1998, ETSI TC MTS started the work on a new version of TTCN. The investigations in this work item studied the strengths and weaknesses of TTCN-2++ and defined the requirements for TTCN-3.

[1] The contents of this section is inspired by presentations from Anthony Wiles and Os Monkewich given at past TTCN-3 user conferences [38,44].

The weaknesses included OSI and CTMF concepts, such as strict layering, PDU, and ASP, that were hardwired into the language. The strengths included the *Point of Control and Observation* (PCO) concept and the support of abstraction from the real test system allowing a sound engineering process with successive abstraction from *Test Purposes* (TPs) via ATS (into TTCN-2 code), to ETS.

Experience with TTCN-2 had also shown the benefits of having a standardized language specifically designed for testing. An abstract testing language allows testers to concentrate on testing rather than on the realization of the test system. A constantly maintained and further developed testing language with commonly understood syntax and operational semantics fosters the development of off-the-shelf tools for test development and test execution. A single language for many test activities reduces the costs for education and test maintenance. Furthermore, such a language allows the application of a common methodology and style, both, on a corporate level, and within standardization.

Based on these findings, ETSI TC MTS started to develop TTCN-3 in 1998. The goal was to base the new testing language on the well-proven concepts of TTCN-2 and to add concepts necessary for emerging testing requirements. TTCN-3 was developed to provide a syntax similar to modern programming languages for testers. It added support for dynamic test configurations and means to enable interactions with the SUT via various communication mechanisms. Powerful concepts for test data and means for flexible control of test case selection, configuration and execution were also provided. In addition, TTCN-3 was harmonized with the data exchange and interface specification mechanisms ASN.1, *Extensible Markup Language* (XML) schema, and *Interface Description Language* (IDL). Furthermore, a TTCN-3-based architecture for automated test execution together with open interfaces and interface mappings to typical programming languages was developed. These concepts enabled an easy interconnection to existing test devices and test platforms.

During the development of TTCN-3, it turned out that the abbreviation *TTCN* was recognized as a brand. However, TTCN-3 no longer includes trees and tables. In order to keep this familiar brand, the abbreviation TTCN was re-interpreted to *Testing and Test Control Notation*. The number 3 in TTCN-3 relates the language to its predecessor TTCN-2.

3 Status of TTCN-3

3.1 TTCN-3 editions

TTCN-3 was first presented to the public in September 2000, and gained the necessary stability and maturity for tool development and its implementation in the industry with edition 2.2.1 in 2003. Subsequent editions that have been issued on an annual basis until now enhanced TTCN-3 with new concepts without breaking backwards compatibility to previous versions. In 2009, TTCN-3 edition 4.1.1 was published and has been since been updated in minor revisions, indicating the reached level of maturity. The latest release of TTCN-3 edition 4.6.1 is planned to be published in 2014. Further major language extensions have been introduced in the TTCN-3 editions 4.x.x by means of extension packages. These packages enable application-specific extensions of TTCN-3 while keeping the core language as comprehensible as possible.

Since 2006, changes to TTCN-3 have been made transparent to the community by means of an issue tracking system [5] set up to implement a *Change Request* (CR) procedure for TTCN-3 (Fig. 1). Over 600 people worldwide have contributed to and/or followed the maintenance process for TTCN-3 in [5]. In addition to the formal system for handling CRs, a more informal mailing list discussion on various topics and features of TTCN-3, involving the broader community has continuously accompanied the standard since its inception.

The article in this special section on “Quantifying the Evolution of TTCN-3 as a Language” by Makedonski et al. [37] looks in detail at the various aspects of the evolution of TTCN-3 from a language and a user perspective. It covers the aspects of maintenance such as text changes in the standard, distribution of CRs across its parts, and activities in the TTCN-3 mailing list, as well as its impact on industrial application and research by an in-depth analysis of the contributions to the various editions of the TTCN-3 user conference series between 2004 and 2011.

3.2 The multi-part standard TTCN-3

As of today, TTCN-3 comprises the following ten parts and six extension packages (Fig. 2):

Part 1: TTCN-3 core language [7]
The specification of the test suites in TTCN-3 can be done in various presentation formats. The textual format of the core language is the most important one as it is of widespread use. The core language serves as a generalized text-based test language in its own right and as the standardized interchange format of TTCN-3 test suites between TTCN-3 tools.

Part 2: TTCN-3 tabular presentation format [9]
The tabular format of TTCN-3 is similar in appearance and functionality to earlier versions of TTCN. It uses a set of tables to denote test suites. It is deprecated and thus no longer maintained.

Part 3: TTCN-3 graphical presentation format [10]
The graphical form of TTCN-3 provides a graphical presentation of the interactions between a test system and an SUT.

Fig. 1 Snapshot of the TTCN-3 online reporting tool in November 2013

Summary

By Project	open	resolved	closed	total
TTCN-3 Change Requests	2	0	0	2
� Ext Pack: Advanced Parametrization (ES 202 784)	2	0	4	6
� Ext Pack: Behaviour Types (ES 202 785)	0	0	4	4
� Ext Pack: Config & Deployment Support (ES 202 781)	14	0	3	17
� Ext Pack: Continuous signal support (ES 202 786)	11	12	1	24
� Ext Pack: Extended TRI (ES 202 789)	0	0	17	17
� Ext Pack: Perf & Real Time Testing (ES 202 782)	3	0	1	4
� Part 01: TTCN-3 Core Language	17	0	393	410
� Part 04: TTCN-3 Operational Semantics	0	0	14	14
� Part 05: TTCN-3 Runtime Interface	0	0	25	25
� Part 06: TTCN-3 Control Interface	1	0	84	85
� Part 07: Using ASN.1 with TTCN-3	0	0	25	25
� Part 08: Using IDL with TTCN-3	0	0	6	6
� Part 09: Using XML with TTCN-3	1	0	88	89
� Part 10: TTCN-3 documentation tags	0	0	12	12

By Status	open	resolved	closed	total
new	1	-	-	1
confirmed	24	-	-	24
assigned	26	-	-	26
resolved	-	12	-	12
closed	-	-	677	677

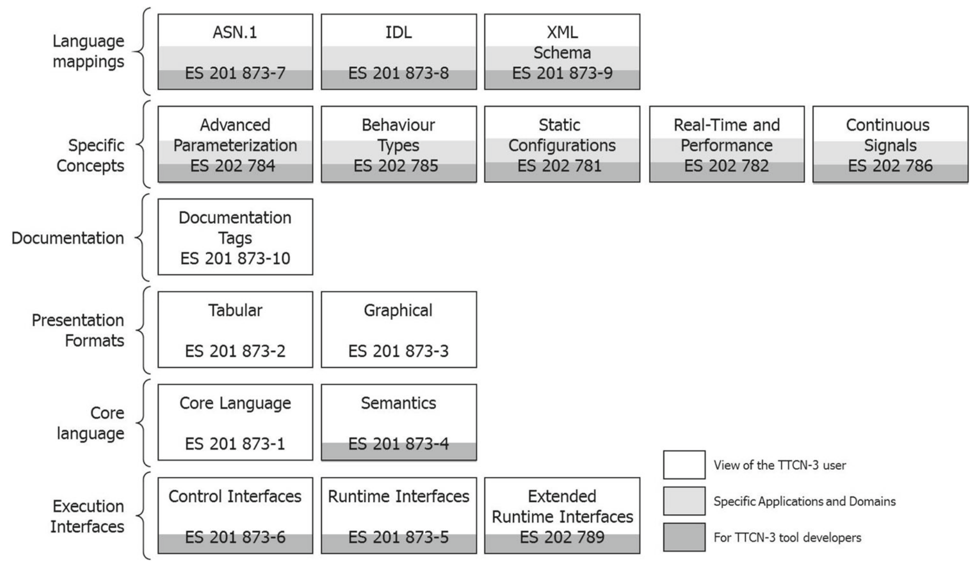
By Severity	open	resolved	closed	total
feature	0	0	50	50
trivial	0	3	16	19
text	0	0	28	28
tweak	0	0	2	2
minor	46	7	486	539
major	5	2	91	98
block	0	0	4	4

By Category	open	resolved	closed	total
Clarification	13	0	191	204
Editorial	7	7	152	166
New Feature	9	0	102	111
Technical	22	5	232	259

It is based on *Message Sequence Charts* (MSCs) and can be used to visualize behavioral aspects of a test specification such as the behavior of a test case or a function. It does not visualize data aspects such as the declaration of types and templates. Similar to the tabular format, the graphical format of TTCN-3 is deprecated and thus no longer maintained.

Part 4: TTCN-3 operational semantics [11]
The operational semantics of TTCN-3 provides the basis for an unambiguous interpretation and execution of TTCN-3 test specifications. It defines the meaning of a test specification in TTCN-3 in an intuitive, yet unambiguous manner and provides a state-oriented view on the execution of a TTCN-3 test specification. The operational semantics is particularly important for tool vendors as it enables semantically identical behaviour of TTCN-3 test specifications, regardless of which TTCN-3 tool is used for execution. However, the operational semantics is provided in a semi-formal way only such that the possibilities to perform formal verification of TTCN-3 test specifications are limited.

Fig. 2 Parts of the TTCN-3 standard series



Part 5: TTCN-3 run-time interfaces [12]
The run-time interfaces of TTCN-3 are one part of the test system architecture of TTCN-3. They define a standardized adaptation layer for the communication and timing interfaces of a test system to the SUT and to a particular execution platform. The run-time interfaces define the required operations independent of a target implementation language in IDL. In addition, mappings to the programming languages C, C++, C#, and Java are provided for ease of implementation.

Part 6: TTCN-3 control interfaces [13]
The control interfaces of TTCN-3 define the second part of the TTCN-3 test system architecture. They provide a standardized adaptation layer for the management of test executions, for the handling of test components in distributed test setups, and for the encoding and decoding of the interactions between the test system and the SUT. The operations of the control interfaces are also specified in IDL together with programming language mappings for C, C++, C#, and Java. Furthermore, the control interfaces provide a standardized logging interface for TTCN-3 test executions. In addition to the programming language mappings, an XML schema for test logs is also provided.

Part 7: Using ASN.1 with TTCN-3 [14]
This part of the TTCN-3 standard defines how ASN.1 can be used for the specification of data types and values in TTCN-3 test descriptions. This enables ASN.1 modules to be imported directly into TTCN-3 modules without additional efforts of converting ASN.1 definitions into TTCN-3 definitions.

Part 8: Using IDL with TTCN-3 [15]
This part of the TTCN-3 standard defines how IDL modules can be imported into TTCN-3 modules, so that system interfaces, operations, and exceptions specified in IDL can be tested directly with TTCN-3 without manual efforts of conversion.

Part 9: Using XML schema with TTCN-3 [16]
This part of the TTCN-3 standard defines how XML schema definitions for data types and values can be used in TTCN-3 test specifications directly without manual efforts for conversion.

Part 10: TTCN-3 documentation comment specification [8]
This part defines means for the documentation of TTCN-3 test specifications using special documentation comments, similar to Javadoc and others. These TTCN-3 documentation tags can be used to produce structured documentation for TTCN-3 test specifications automatically, for example, in the form of hypertext web pages.

In addition to the main parts of the TTCN-3 standard, there are currently six extension packages of TTCN-3. The intention of the TTCN-3 extension packages is to support additional concepts required by specific TTCN-3 application domains, without making them mandatory for all test suites or tool environments. Rather, these packages can be combined with the TTCN-3 core language as needed, depending on the specific requirements of a given application domain. For example, a tool environment for automotive electronic control units may require the real-time package of TTCN-3, but not the behavior types package. As of today, the following TTCN-3 extension packages have been standardized.

Extension Package: TTCN-3 configuration and deployment support [18]
The configuration and deployment support language extension defines the concept of static test configuration, as well as its semantics and execution details. A static test configu-

ration is a test configuration with a lifetime that is not bound to a single test case, but can be used across several test cases. The static test configuration continues to exist and keeps its configuration and parametrization after the termination of a test case, such that a subsequent test case can be executed on the same test configuration as left by the previous test case.

Extension Package: TTCN-3 performance and real-time testing [19]
The extension package on performance and real-time testing provides the means for the measurement of time, for the specification of time points and time spans, for the precise timing of the stimulation, and for the calculation and comparison of time values. Moreover, the test execution architecture of TTCN-3 is extended so that a TTCN-3 engine can ensure that the specified time-related actions (time measurement, timed stimulation) are executed correctly with respect to the required precision.

Extension Package: TTCN-3 advanced parameterization [20]
The advanced parameterization extension package provides means for the parameterizing of TTCN-3 types with type and value parameters. This enables more flexibility in the use and application of types in TTCN-3 test specifications.

Extension Package: TTCN-3 behavior types [21]
The extension package on behavior types introduces behavior types for TTCN-3 in order to increase the re-usability of test behavior definitions. Behavior types denote altsteps, functions, and test cases of a TTCN-3 test specification that have compatible parameter lists, return types, and compatible runs on or system clauses.

Extension Package: TTCN-3 support of interfaces with continuous signals [22]
The continuous signal extension package of TTCN-3 defines the concept of continuous signals in TTCN-3. Continuous signals describe progressive series of values of a type at every instant of time or at a given interval, expressed by means of streams. For the definition and evaluation of continuous signals the concept of a *mode* is introduced, which characterizes a defined periodic activity of a test system or SUT component over time. Furthermore, continuous signals can be processed post-mortem, i.e., after a test run, as history-traces. Mathematical functions can be used to analyze these history-traces.
The article in this special section on "Testing Hybrid Systems with *TTCN-3 embedded*" by Grossmann [28] provides an overview of the language concepts for this TTCN-3 extension that has particular value for testing embedded devices in the automotive and avionics domains that communicate with the physical world via sensors and actuators. The concept of continuous signals is captured in the notion of a *hybrid system* that exhibits continuous and discrete behavior. Since core TTCN-3 deals only with the exchange of discrete interactions between the tester and the SUT, i.e., messages or API calls, the article describes the necessary language extension on a sound semantic basis and discusses the new concepts on an automotive case study.

Extension Package: TTCN-3 extended TRI [23]
The extension package on an extended version for the runtime interfaces of TTCN-3 provides a more efficient handling of software objects in an interaction with an SUT. It is particularly useful for such SUTs that do not use binary encoded messages in interactions, but rather the serialized values. In such cases software objects can be passed directly between the SUT and the test system.

The article in this special section on "Innovation and Evolution in Integrated Web Application Testing with TTCN-3" by Stepien and Peyton [43] highlights this particular aspect of testing object-oriented systems. The work by these authors influenced the development of this extension package. Specifically, the authors consider web applications as their test target and reason about a streamlined and simplified TTCN-3 test architecture that bypasses the explicit integration of a codec for message encoding/decoding and enables the integration of TTCN-3 within xUnit test frameworks such as JUnit [35] and HtmlUnit [29].

3.3 Tooling for TTCN-3

Numerous of tools and auxiliary tool components and test devices exist that implement TTCN-3.

- Test development and test execution environments based, e.g., on Eclipse, which support the development and execution of TTCN-3 test specifications;
- Test management tools with integrated TTCN-3 support;
- Test adapters, e.g., for TCP/IP socket communication, serial interfaces, etc., to support the adaptation of TTCN-3 test executables to various SUTs;
- Test devices, e.g., for *Universal Mobile Telecommunications System* (UMTS), *Long Term Evolution* (LTE) *Integrating the Healthcare Enterprise* (IHE), etc.;
- Test generators to derive TTCN-3 test specifications from the *Unified Modeling Language* (UML) models or other modeling languages;
- Test documentation tools to generate documentations for test specification, test traces and test reports;
- Test quality tools to measure, validate, and optimize the correctness, maintainability, and/or readability of TTCN-3 specifications; and
- Executable test suites, e.g., for *Session Initiated Protocol* (SIP), *Internet Protocol version 6* (IPv6), and alike, which are based on TTCN-3 test specifications.

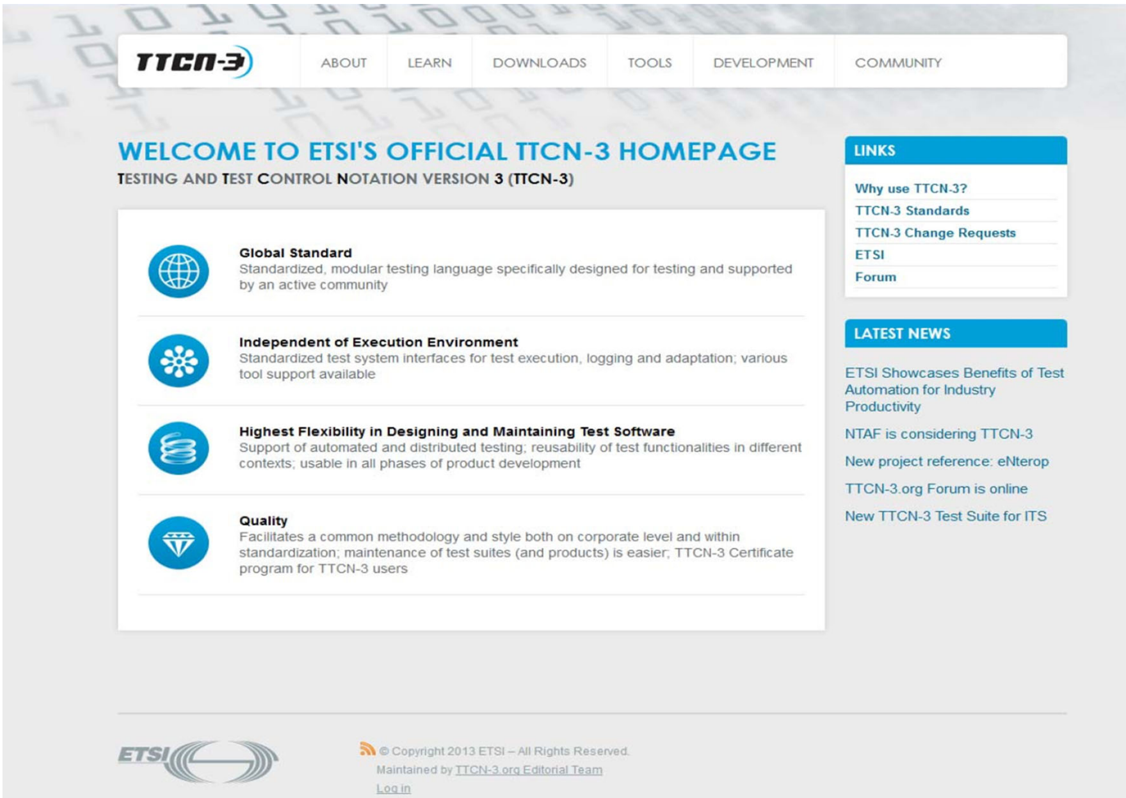


Fig. 3 TTCN-3 web page at [6]

The majority of the tools is commercial. Some tools are provided as open source projects or as research prototypes. More details on tooling are available at [6] (Fig. 3).

An important aspect in this context is the availability of a TTCN-3 conformance test suite in TTCN-3 for testing the conformance of a TTCN-3 tool against the TTCN-3 standard itself. Given the complexity of the language, different tools such as compilers might interpret the standard differently, which could result in TTCN-3 test suites that execute differently in different tool environments or are different even at the syntactical level.

The article in this special section on "A Conformance Test Suite for TTCN-3 Tools" by Zeiss et al. [45] discusses the approach chosen to derive conformance tests for tools in terms of generated TTCN-3 ATS. In addition, it describes a methodology to execute a generated abstract test suite and reason about the test outcome from executing such tests. While the article lays down the foundation for defining TTCN-3 conformance tests, this activity continues until today to reach sufficient coverage of the standard.

3.4 The use of TTCN-3 in standardization

Since TTCN-3 is the only standardized test technology which provides all concepts needed to specify test suites completely, i.e., in a manner that the test suites can be automatically executed, it is a test technology that is applied by other standardization bodies to specify conformance and interoperability tests. This is particularly true for ETSI as the owner, developer, and maintainer of TTCN-3, but also other standardization bodies such as the *3rd Generation Partnership Project* (3GPP), the *Automotive Open System Architecture* (AUTOSAR), the *Terrestrial Trunked Radio* (TETRA), and the Open Mobile Access (OMA) applies TTCN-3 to offer standardized test suites. The ETSI test specifications in TTCN-3 cover tests for:

- Worldwide Interoperability for Microwave Access (WiMax) (802.16);
- ePassport readers interoperability;
- Voice over Internet Protocol (IP) with SIP;
- IP Multimedia Subsystem (IMS);
- IMS interworking;
- IPv6;
- Digital Private Mobile Radio (dPMR);
- Digital Mobile Radio (DMR); and
- Intelligent Transportation Systems (ITS).

A number of large deployments on test devices contain TTCN-3 test specifications of 3GPP standards that include tests for:

- evolved UMTS Terrestrial Radio Access (e-UTRA) (LTE/EPC) User Equipment (UE);
- IMS UE;
- UMTS Terrestrial Radio Access (UTRA) UE; and
- UE positioning.

Last but not least, there are TTCN-3 libraries available that help in the establishment or further development of test suites in selected domains. They include a common library for distributed test configurations and dedicated libraries for IMS, ITS, SIP, and others.

The article in this special section on “A Generic Interoperability Testing Framework and a Systematic Development Process for Automated Interoperability Testing” by Rings et al. [41] represents a comprehensive piece of work to support this type of testing for IMS implementations, which make up an important share of today’s mobile network infrastructure. The approach is based on a mixture of checking end-to-end functionality and validating data observed at additional test probes. It is implemented in interoperability events organized by ETSI that feature tests of network equipment provided by different vendors. Although the presented concepts of interoperability testing aim certainly at fulfilling ETSI’s requirements on such tests, they are described independently from the concrete application such that they can be reused to solve similar testing tasks.

3.5 Education for TTCN-3

Last but not least, the TTCN-3 community has established a training program since 2007 to educate testers in designing, specifying, and implementing test suites. Training courses built on the basis of this syllabus aim at all people involved in software testing, who wish to get acquainted with and gather experience in TTCN-3, such as quality and test professionals, programmers, developers, specialists, and (project-) managers responsible for executing, planning or controling testing.

The objectives of the training courses attempt to make TTCN-3-Certificate holders well-acquainted with TTCN-3 concepts, enable them to practically apply and integrate TTCN-3 into established test processes, and build test processes on the basis of TTCN-3. The training courses comprise concepts, presentation formats, guidelines of usage and application examples:

- TTCN-3-Certificate holders are capable of designing and planning the project-specific tests and formalize them in TTCN-3. They are capable of defining adequate test strategies including test objectives, TPs, test configurations, test suites, and the realization of tests on target test platforms.
- TTCN-3-Certificate holders are capable of developing functional, conformance, interoperability, robustness, scalability, load, stress, etc., tests in TTCN-3.
- TTCN-3-Certificate holders are aware that TTCN-3 is a powerful test technology, but not the only one. They are capable of identifying those test targets which gain most from a systematic, formalized test approach with TTCN-3. They are capable of developing mixed test strategies (choosing from a set of test technologies) to have optimized test efficiency.
- TTCN-3-Certificate holders are capable of putting TTCN-3 tests into practice. They are capable of setting up automated tool chains for the execution of tests, creation of corresponding test reports, and the interaction with the system developers.
- TTCN-3-Certificate holders are able to develop TTCN-3 based testing processes and to integrate the testing process with the system development process.

The TTCN-3 education program is hosted by the *German Testing Board* (GTB). Both, ETSI and *International Software Quality Institute* (iSQI) serve as certification bodies. There is currently just one accredited training provider, but the program is open also for others.

4 The Future of TTCN-3

4.1 Are the times changing, again?

TTCN-3 has evolved from its first edition in 2000 to an elaborated and comprehensive test methodology. Similar to TTCN-2 in 1999, questions arise concerning the suitability of TTCN-3 for coping with the challenges of testing upcoming new technologies and applications.

In the last decade, there has been an ever increasing influence of sophisticated Internet services in people’s personal and professional daily lives. The usage of online services for vital operating procedures such as banking, shopping, tax declarations, or contract insurances has become common practice. In addition, more and more safety-critical systems including cars, trains, aircraft, medical devices, and many others require efficient and reliable software systems and infrastructures.

ETSI and the TTCN-3 community have addressed the technical challenges of these and other new technologies by adding new core language features and by defining extension packages tailored to specific application areas. For the user, TTCN-3 extensions augment the core language and provide a test implementation view which often is close to test execution.

4.2 What about abstraction?

Even though TTCN-3 provides a tabular and a graphical presentation format for test specifications, it is rarely used for test specifications at higher levels of abstraction, e.g., at the level of TPs. One reason is that TTCN-3 provides a vertical (implementation) view on test behavior, i.e., test cases are specified by defining the (detailed) behavior of the test components. The disadvantage of this approach is that communication among the test components and with the SUT is difficult to track. Thus, test specifiers prefer notations like UML Sequence Diagrams [26] that provide a scenario and information flow-oriented view for high-level test designs.

ETSI uses the *Test Purpose Notation* (TPLan) [17] for the specification of TPs in its test standardization process. TPLan provides a (rough) frame and a set of keywords with a test-specific meaning enabling the description of test cases in structured English. Currently, MTS attempts to further formalize high-level test specification by means of the *Test Description Language* (TDL) [1].

In order to cope with the increasing complexity of test specifications for upcoming systems, ETSI also has investigated the usage of *Model-Based Testing* (MBT) technologies in test suite standardization [27]. Case studies have shown the applicability of MBT technologies in the standardization process. However, ETSI has an elaborated document-based standardization process where TPs are discussed individually. Currently, there are only initial proposals for the establishment of an engineering process for test modeling and the integration of this process into test standardization.

Even though abstraction matters, the role of TTCN-3 for the description of tests close to implementation and execution is not put into question. TTCN-3 has found its place in test standardization at this level of abstraction.

4.3 What about competitors?

Test support at the abstraction level of implementation languages is provided by various XUnit frameworks, such as JUnit [35], NUnit [39], and CppUnit [4]. However, the XUnit frameworks are tailored to unit tests developed and executed by programmers, whereas TTCN-3 addresses higher testing levels such as integration, system, acceptance, conformance, and interoperability testing. Even though TTCN-3 may also be used for unit testing, it cannot compete with the XUnit frameworks. The advantage of the XUnit frameworks is the integration of testing concepts into the used programming language. Developers can apply the framework without learning a new language. At higher test levels, the XUnit frameworks cannot compete with TTCN-3, though, as these test levels require abstraction from specific programming languages and environments.

The *UML Testing Profile* (UTP) [3,40] is the TTCN-3 competitor from the UML world. UTP extends UML with testing concepts. Thus, system design and test specification can be done at the same level of abstraction. Interestingly, the competition between UTP and TTCN-3 is rather small. The UML community considers TTCN-3 as a test implementation and test execution language, whereas TTCN-3 users seem to not require the comprehensive modeling capabilities provided by UTP.

A further competitor to TTCN-3 may evolve from the *Automatic Test Markup Language* (ATML) standardized by the IEEE 1641 standards series [2]. ATML enables interoperability among test systems by standardizing an XML-based information exchange of test descriptions, test configurations, test adapters, test results, and test stations. In contrast to ATML, TTCN-3 has been developed for test specifications and test implementations. Interoperability is considered to be important only with respect to TTCN-3-based test systems and this interoperability is assured by providing TTCN-3 specific run-time interfaces [12,13]. ATML may evolve into a serious TTCN-3 competitor if it adopts a test design language similar to the TTCN-3 core language.

4.4 And the TTCN-3 users?

It is difficult to estimate the number of TTCN-3 users, but several indicators show the substantial and continued user interest in TTCN-3. Currently, the TTCN-3 web site [6] lists six commercial TTCN-3 tool vendors. In addition, company-internal tools have been developed and are continuously maintained. Most companies provide not only tools, but also offer training courses and consultancy services. Thus, TTCN-3 is commercially relevant.

In the TTCN-3 maintenance process, even after 10 years of maintenance, the number of CRs is not decreasing. However, the increasing number of small issues (e.g., requests for clarification or for the removal of ambiguities in the standard texts) reported for the TTCN-3 core language indicate an increasing stability of the standard. Even after more than 10 years of development, the TTCN-3 standard is still growing. Each year proposals for new constructs and extension packages are submitted to the TTCN-3 online reporting system or are discussed within ETSI TC MTS. Recently, language concepts for supporting security testing by means of fuzzing have been proposed [42].

Until 2011, TTCN-3 had its own user conference [2], where practitioners and researchers presented their experiences and research on testing with TTCN-3. Starting with 2013, the TTCN-3 user conference series was merged with the broader-scoped *User Conference on Advanced Automated Testing* (UCAAT) [25], which incorporates further experiences and research in automated testing, including MBT. UCAAT 2013 still attracted a number of presentations on TTCN-3.

[2] A list of all former TTCN-3 user conferences events together with their given presentations can be found at http://www.ttcn-3.org/index.php/community/events.

5 Conclusions

In this article, the history and status of TTCN-3 was presented. It was described how TTCN-3 evolved from its roots in OSI conformance testing to a mature test technology applicable to a broad range of applications and to various kinds of testing. TTCN-3 is frequently used in industry and standardization as a test development and test specification technique that is close to test implementation and test execution. In this application area, TTCN-3 hardly has any competitors.

Even after more than 10 years of development, TTCN-3 is still evolving. The language is constantly maintained and users continue to request new features. There are no signs of decreasing interest in TTCN-3, but there are also no signs of a new TTCN-3 boom.

From the authors' point of view, the TTCN-3 concepts heavily influenced the testing world in the last decade. Even though TTCN-3 is not always used directly, the concepts of TTCN-3 influenced other test technologies, tools and solutions.

For the future, more research on high-level test design approaches and on test automation is expected. TTCN-3 will likely continue to contribute to both. On the one hand, concepts identified in TTCN-3 will also be needed for high-level test designs. On the other hand, TTCN-3 already increased the understanding of the fundamental concepts of test automation by separating test design from concrete test implementations. In addition to the automated generation and execution of tests, future test technologies may also support the automated generation of SUT- and/or test system-specific test frameworks. TTCN-3 provides a solid starting point for these future pursuits.

Articles selected for this special section

The following five articles of this special section have been already presented in this article. They provide further details of the evolution and some recent trends in the further development of TTCN-3:

- The article "Quantifying the Evolution of TTCN-3 as a Language" by Makedonski et al. [37] investigates various aspects of the TTCN-3 evolution.
- The article "Testing Hybrid Systems with *TTCN-3 embedded*" by Grossmann [28] presents concepts and background of the extension Package "TTCN-3 Support of Interfaces with Continuous Signals" [22].
- Particular aspects of the testing of object-oriented systems are highlighted in the article "Innovation and Evolution in Integrated Web Application Testing with TTCN-3" by Stepien and Peyton [43].
- The unique approach of providing conformance tests for TTCN-3 tools by means of TTCN-3 test suites is discussed in the article "A Conformance Test Suite for TTCN-3 Tools" by Zeiss et al. [45].
- The role and usage of TTCN-3 for automated interoperability testing is studied in the article "A Generic Interoperability Testing Framework and a Systematic Development Process for Automated Interoperability Testing" by Rings et al. [41].

Acknowledgments The editors of the special section wish to express their gratitude to all authors, reviewers, and the STTT editorial team for their contributions, help, and patience during the composition of this Special Section.

References

1. Adamis, G., Kristoffersen, F., Makedonski, P., Ulrich, A., Wendland, M.: An overview of the ETSI Test Description Language (TDL)—Results from STF 454. Presentation at the first ETSI User Conference on Advanced Automated Testing (UCAAT'13), Paris, 22–24 Oct 2013. http://ucaat.etsi.org/2013/presentations/Intro%20to%20ETSI%20TDL_Andreas%20Ulrich.pdf. Accessed 06 Jan 2014
2. ATML Hompage: http://grouper.ieee.org/groups/scc20/tii/. Accessed 06 Jan 2014
3. Baker, P., Dai, Z., Grabowski, J., Haugen, O., Schieferdecker, I., Williams, C.: Model-Driven Testing—Using the UML Testing Profile. Springer, Berlin Heidelberg (2007)
4. CppUnit Hompage: http://sourceforge.net/projects/cppunit/. Accessed 06 Jan 2014
5. ETSI: TTCN-3 Change Request System: http://t-ort.etsi.org/. Accessed 06 Jan 2014
6. ETSI: TTCN-3 Web Site; http://www.ttcn-3.org/. Accessed 06 Jan 2014
7. ETSI ES 201 873–1: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 1: Core Language, v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
8. ETSI ES 201 873–10: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 10: TTCN-3 Documentation Comment Specification, v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
9. ETSI ES 201 873–2: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 2: Tabular Presentation Format (TFT), v3.2.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2007)
10. ETSI ES 201 873–3: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 3: Graphical Presentation Format (GFT), v3.2.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2007)
11. ETSI ES 201 873–4: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 4: Operational Semantics, v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
12. ETSI ES 201 873–5: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 5: Run-Time Interface (TRI), v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)

13. ETSI ES 201 873–6: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 6: Control Interface (TCI), v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
14. ETSI ES 201 873–7: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 7: Using ASN.1 with TTCN-3, v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
15. ETSI ES 201 873–8: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 8: Using IDL with TTCN-3, v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
16. ETSI ES 201 873–9: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3;—Part 9: Using XML with TTCN-3, v4.5.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
17. ETSI ES 202 553: Methods for Testing and Specification (MTS); TPLan: A notation for expressing Test Purposes, v1.2.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2009)
18. ETSI ES 202 781: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Extensions: Configuration and Deployment Support, v1.2.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
19. ETSI ES 202 782: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Extensions: Performance and Real-time Testing, v1.1.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2010)
20. ETSI ES 202 784: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Extensions: Advanced Parametrization, v1.3.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
21. ETSI ES 202 785: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Extensions: Behaviour Types, v1.3.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
22. ETSI ES 202 786: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Extensions: Support of Interfaces with Continuous Signals, v1.1.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2012)
23. ETSI ES 202 789: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; TTCN-3 Extensions: Extended TRI, v1.2.1. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (2013)
24. ETSI TR 101 666: Information technology — Open Systems Interconnection Conformance Testing Methodology and Framework; The Tree and Tabular Combined Notation (TTCN) (Ed. 2++). European Telecommunications Standards Institute (ETSI), Sophia-Antipolis (1999)
25. First ETSI User Conference on Advanced Automated Testing (UCAAT'13), Paris, 22–24 October 2013. http://ucaat.etsi.org/2013/. Accessed 06 Jan 2014
26. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd edn. Addison-Wesley Professional (2003)
27. Grabowski, J., Kuliamin, V., Vouffo Feudjio, A., Wu-Hen-Chang, A., Zoric, M.: Towards the Usage of MBT at ETSI. In: MBT 2013: proceedings of the eighth workshop on Model-Based Testing (MBT 2013), Rome, 17th Mar 2013, electronic proceedings in theoretical computer science, vol 111, pp. 30–34. http://arxiv.org/abs/1303.1007. Accessed 06 Jan 2014 (2013)
28. Grossmann, J.: Testing hybrid systems with TTCN-3 embedded. Int. J. Software Tools Technol. Trans. (STTT), ISSN 1433–2779 (2013). doi:10.1007/s10009-013-0283-0
29. HtmlUnit Hompage: http://htmlunit.sourceforge.net/. Accessed 23 Jan 2014
30. ISO/IEC: Information processing systems — Open Systems Interconnection—LOTOS—A formal description technique based on the temporal ordering of observational behaviour. International ISO/IEC standard No. 8807 (1989)
31. ISO/IEC: Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model. International ISO/IEC standard No. 7498–1 (1994)
32. ISO/IEC: Information technology—Open Systems Interconnection—Conformance testing methodology and framework. International ISO/IEC multipart standard No. 9646 (1994–1998)
33. ISO/IEC: Information technology—Open Systems Interconnection—Conformance testing methodology and framework—Part 3: The Tree and Tabular Combined Notation (TTCN). International ISO/IEC standard No. 9646–3 (1998)
34. ITU-T: Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation. ITU-T recommendation X.680 (11/2008) (2008)
35. JUnit Hompage: http://junit.org/. Accessed 06 Jan 2014
36. Kroon, J., Wiles, A.: A Tutorial on TTCN. Tutorial at the 11th International IFIP WG6.1 symposium on Protocol, Specification, Testing and Verification (PSTV) (1991)
37. Makedonski, P., Grabowski, J., Philipp, F.: Quantifying the evolution of TTCN-3 as a language. Int. J. Software Tools Technol. Trans. (STTT), ISSN 1433–2779 (2013). doi:10.1007/s10009-013-0282-1
38. Monkewich, O.: Ten Years of TTCN-3—Past, Present and Future. Presentation at the ETSI TTCN-3 User Conference 2011 (T3UC'11), Bled, Slovenia, 7–9 June 2011. http://www.ttcn-3.org/TTCN3UC2011/Pres/08_T3UC-Monkewich-TenYearsOfTTCN3-PastAndFuture.ppt. Accessed 06 Jan 2014
39. NUnit Hompage: http://www.nunit.org/. Accessed 06 Jan 2014
40. Object Management Group (OMG): UML Testing Profile (UTP), Version 1.2. OMG Document Number: formal/2013-04-03, Standard document http://www.omg.org/spec/UTP/1.2/ (2013)
41. Rings, T., Poglitsch, P., Schulz, S., Serazio, L., Vassiliou-Gioles, T.: A generic interoperability testing framework and a systematic development process for automated interoperability testing. Int. J. Software Tools Technol. Trans. (STTT), ISSN 1433–2779 (2013). doi:10.1007/s10009-013-0281-2
42. Schneider, M., Grossmann, J., Schieferdecker, I., Pietschker, A.: Online model-based behavioral fuzzing. ICSTW'13: proceedings of the 2013 IEEE sixth International Conference on Software Testing Verification and Validation, Workshops, pp. 469–475. IEEE Computer Society, Washington, DC (2013)
43. Stepien, B., Peyton, L.: Innovation and evolution in integrated web application testing with TTCN-3. Int. J. Software Tools Technol. Trans. (STTT), ISSN 1433–2779 (2013). doi:10.1007/s10009-013-0278-x
44. Wiles, A.: The History and Future of TTCN-3. Presentation at the ETSI TTCN-3 User Conference 2007 (T3UC'07), Stockholm, 29 May–1 June 2007. http://www.ttcn-3.org/TTCN3UC2007/Presentations/Thu/ETSI%20TTCN-3%20keynote.pdf. Accessed 06 Jan 2014
45. Zeiss, B., Kovacs, A., Pakulin, N., Stanca-Kaposta, B.: A conformance test suite for TTCN-3 tools. Int. J. Software Tools Technol. Trans. (STTT), ISSN 1433–2779 (2013) doi:10.1007/s10009-013-0285-y