

Use of TTCN-3 for the Development of SIGTRAN Tests

Dimitrios Apostolidis, Dirk Tepelmann

Testing Technologies IST GmbH
Rosenthaler Str. 13
10119 Berlin, Germany
www.testingtech.de

Axel Rennoch, Alain Vouffo

Fraunhofer Fokus, CC Motion
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
www.fokus.fraunhofer.de/motion

Abstract: SIGTRAN refers to the signaling transport protocol stack standardized by IETF to build a bridge between SS7 and IP based signaling systems. Since various SIGTRAN recommendations are new and still under development there is an actual need for testing related products. The authors have investigated in the development of TTCN-3 test suites for the SIGTRAN protocols M3UA and SUA.

The paper gives an insight on the development work of large industrial TTCN-3 tests. The presentation explains the selected design approach, including test configuration or module parameters. Furthermore, we describe our experiences w.r.t. the definition of (parameterized) data templates and behavior sequences. The challenges and efforts due to the test suite validation and execution will be discussed, too.

Keywords: Testing, ATS, TTCN-3, M3UA, SUA

1. INTRODUCTION

TTCN-3 as the only international test language standard [CN] has been explained very well in articles and books (see e.g. [W05]). It has been accepted in several international projects, performed through e.g. ETSI (SIP tests [W01]) or the WiMAX-Forum. Although the Conformance Testing Methodologies have been defined in principle and standardized since the beginning of the nineties [CTMF] and a lot of test suites have been produced and executed the Abstract test suite (ATS) development process still need experts who are familiar with both testing and the SUT target technologies. Furthermore the newest version of the test notation TTCN-3 and related tools provide the means to innovate the production process to make it more efficient and cheaper.

Even though there are multiple TTCN-3 code examples in the language introductions there is still need to explain the setup of a TTCN-3 project that target is the production of an TTCN-3 based industrial test suite ready to execute against a System under test (SUT). A series of questions have to be answered before the production of the test cases could start: e.g. a suitable test configuration, the types of the ports etc.

Our experience report addresses test suite development for the SIGTRAN protocol stack as defined in [SIGT]. Signaling gateways have been introduced to bridge signaling message (information) from a switched circuit network into an IP based network. In this study we focus on the user adaptation protocols [M3UA] and [SUA] that provide signaling service access points according to MTP layer 3 and SCCP resp., as depicted in Figure 1. Both protocols are located above the [SCTP] layer and do connect signaling gateway and application service processes.

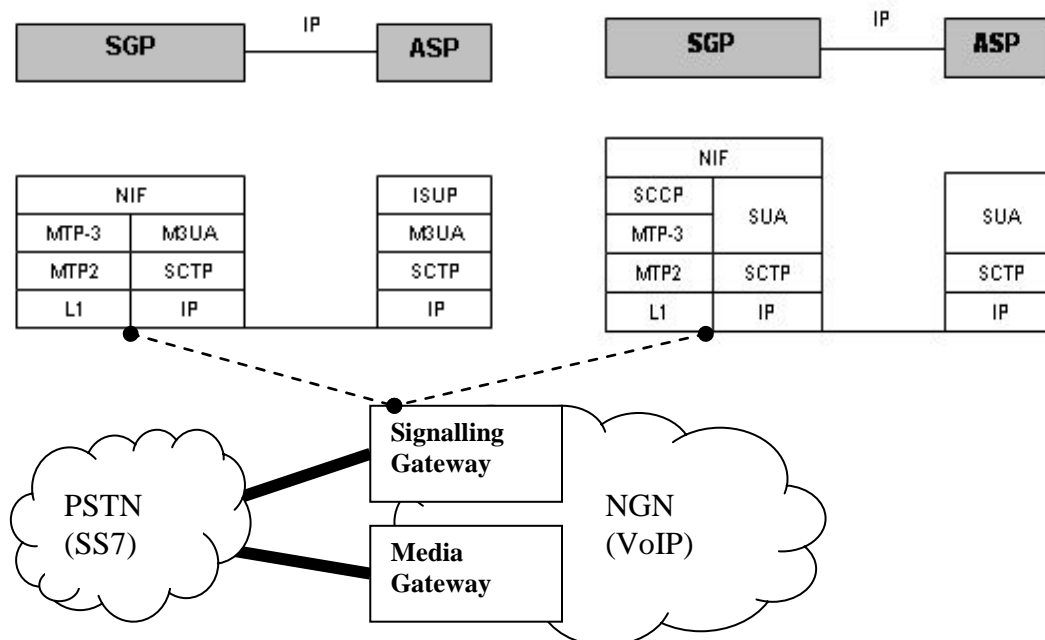


Figure 1: M3UA and SUA protocol stacks

In the following section 2 we present an overview on the ATS development process and details collected during the development of the ATS focusing on different topics as configuration, data and behavior definitions and ATS parameters. In section 3 we give some information on the test system adapter and codec that has to be implemented to execute the test suite. Furthermore we explain in section 4 our approaches for validating the tests and in section 5 the experiences and results from a project management viewpoint.

2. ATS DEVELOPMENT

Today in most cases there is no formal (i.e. machine processable) system model of the SUT available and time pressure doesn't allow developing any formal model to get an automatic tool support for test suite derivation. Thus a manual-written synthesis of the test suite is required. In this section we assume the availability of an existing test plan, i.e. a test suite structure and test purposes (TSS & TP) document that has to be considered for ATS development.

In the industrial environment the TTCN-3 test suite development process follows a clear sequence of development steps: As illustrated in Figure 2 at the beginning a common set of naming conventions within the project, the finding of suitable test system interfaces and configurations for the selected test architectures and a structure for the TTCN-3 modules will be addressed. In a second step the data type system has to be specified. Existing libraries on type definitions (TTCN-3 or other languages like ASN.1, IDL or XML) or frameworks with reusable TTCN-3 functions (e.g. [D04]) have to be considered. At this point it is also the time to agree on potential auxiliary logging features enabling the automatic production of meaningful test reports during test campaigns.

After the clarification of the above issues a prototype test scenario with preamble (pr), testbody (tb), postamble (po) and default (def) behavior should be written and evaluated before the mass of the test cases will be produced. System adapter (SA), codec (CD) and external functions could be implemented in parallel to the last two steps.

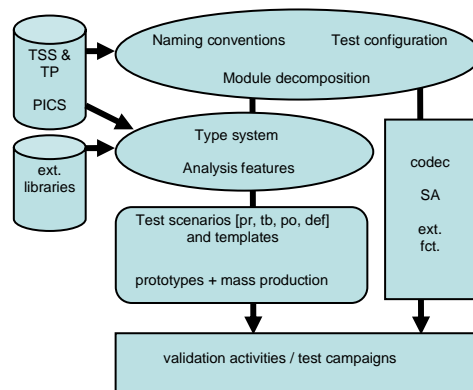


Figure 2: Test development process

There are no standard naming conventions available and even in different ETSI projects on TTCN-3 test suites different naming conventions have been found [IPv6][SIP]. The latter situation is due to individual preferences and existing naming conventions coming with the SUT standards. Interworking test suites may even apply different naming conventions for two involved protocols in order to reuse existing definitions for one of the protocols. Nevertheless an agreement on prefixes for e.g. functions, defaults, templates are recommended.

2.1 Module decomposition

Since all large TTCN-3 projects result in huge source code files at the very beginning a clear ATS file structure should be established to separate different concerns. A sample structure is proposed in the following:

- x_ATS.ttcn3, separating major test case groups,
- x_ASP.ttcn3, abstract service primitives (if needed),
- x_Data.ttcn3, including data templates,
- x_Settings.ttcn3, test suite parameters,
- x_TestCasesY.ttcn3, containing one or more test case groups, and
- x_Types.ttcn3, for the protocol data types,
- x_ConfigAndFunction.ttcn3, for the component, port and function definitions,
- x_Reasons.ttcn3, text strings used, error message logs,

where x is a placeholder for the protocol identifier (e.g. M3UA). TTCN-3 import statements have to be used to ensure the completeness of the specification parts during compiler checks. See Figure 3 for the import relationships.

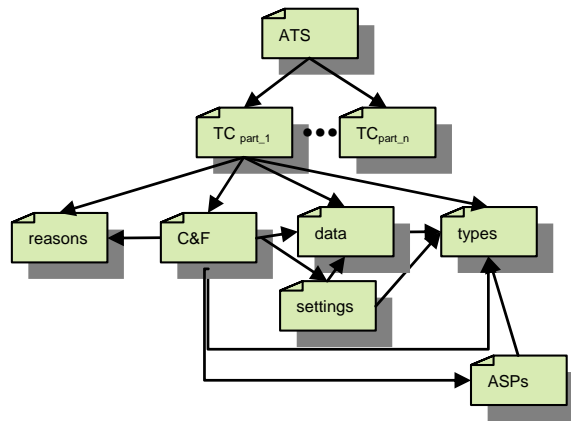


Figure 3: ATS Filestructure

2.2 Configuration

The most important question to be answered before writing test cases address the test configuration, i.e. number of required test components and the definitions of port types they will support. At this point the theoretical accessibility of the SUT have to be known very well, in particular the kind of involved service access points and management interfaces are essential, too. The concepts of a particular SUT implementation help to design the test configuration but also it should be clear that the ATS must be independent of a specific SUT realization to be applicable for all possible implementations of a protocol standard.

In protocol testing there is always an underlying communication service that delivers the PDUs between protocol entities. Although here a message-based port for SUT messages is quite obvious it is also reasonable to specify a procedure-based port for this exchange. The latter case allows the introduction of TTCN-3 operations, e.g. for the control of the communication service (reset, etc.), and PDUs from/to the SUT could be handled as parameters of data transport primitives.

In order to develop a comprehensive ATS all possible interaction points have to be planned. Missing upper tester interfaces may require the consideration of different auxiliary interfaces in order to trigger the SUT initiated tests (e.g. using UNI during NNI testing [D98]). Management interface activities may appear in test purpose definition even if there is no exact information available about management primitives. As long as there is not enough information available about an interface of the SUT in our projects we follow the approach to assume communication channels for the exchange of simple data packets (using record structures). These ports are always message-based.

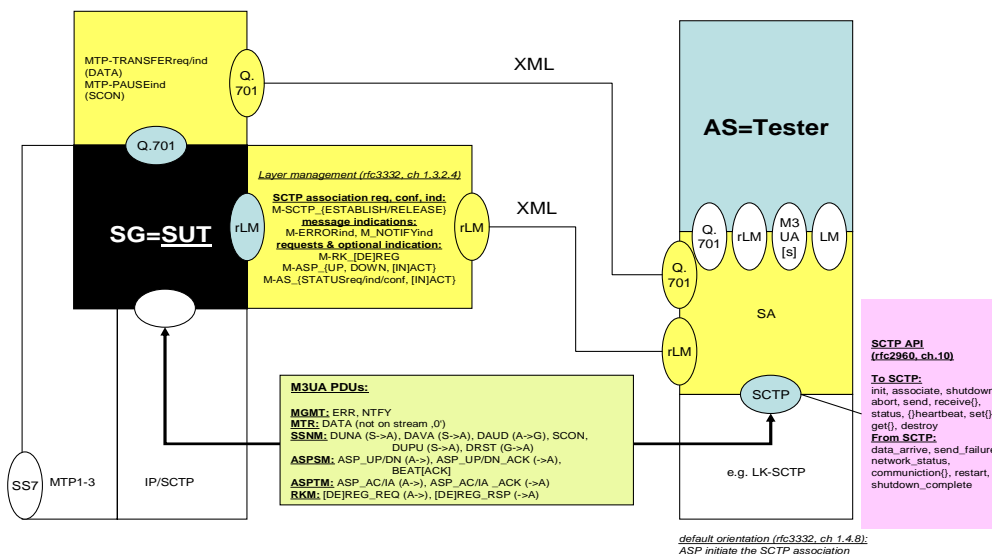


Figure 4: M3UA test configuration for testing a SGP

Our approach in the M3UA project (which is similar to the SUA configuration) is given in Figure 4 that illustrates a principle test configuration for testing the SUT at the Signaling Gateway (SG) side. From the testing viewpoint (to be defined in the ATS) the following different communication ports are identified and assumed to observe and control the SUT:

- M3UA is used for the exchange of M3UA messages. An index *s* has been introduced to distinguish the different stream numbers of the underlying SCTP association.
- rLM is introduced for the interaction with the remote Layer management.
- rMTP3user (according to Q.701) represents the MTP layer 3 service access point for any MTP3 user above the M3UA layer under test, i.e. it is the upper tester interface.
- LM is of local (test system) interest only and provides a means to initiate actions at the local SCTP endpoint. Furthermore it is used by the test adapter to indicate relevant events which has occurred w.r.t. SCTP associations.

M3UA messages are delivered via the underlying SCTP layer. rLM and rMTP3user ports are prepared to allow an interaction of the tester from the M3UA user perspective (upper tester) and with layer management of the SUT. In both cases a (XML) message based communication is prepared for the information exchange.

Several test configurations assume more than one SCTP association for the interaction with the SUT. Distinct SCTP associations are used if different ASPs or SGP are involved. Up to four different SCTP associations can be distinguished according to the actual test configurations: M3UA, M3UA_2, M3UA_3 and M3UA_4. Different streams of SCTP associations are distinguished using the TTCN-3 “array of port” concept, e.g. M3UA[i] represents stream “i” of the association.

2.3 Data type and template definitions

SIGTRAN message data types could be defined easily from the original documents since pattern formats are given in the recommendations for the common message header and each of the parameter structures (using TTCN-3 record data types). Simple TLV (tag-length-value) parameters with a single octet string value refer to a generic parameter structure only, but others need their individual structure definitions.

For each message type a particular TTCN-3 “set” data type contains the list of allowed parameters. If an invalid message type (e.g. containing invalid parameters) is required due to a particular test purpose a separate message type must be specified. A special optional “UnknownParameters” set (of parameters) has been included for each message type due to message parameters that may not be identified by the test system.

One of the most extensive ATS parts are data templates. Following the type definitions we distinguish between templates for message headers and bodies. All header templates are parameterized by an integer length value which is transformed to a four octet value before its assignment to the message length element. Each individual message to be sent or received has an identifier that informs about the message type, the direction (s or r) and an index (messages with non-conformant contents get an additional ‘i’), e.g. “duna_s_i_3” identifies an invalid DUNA message.

There are no signature definitions in the ATS even for the service interfaces explained in the configuration. ASP types and related template are introduced to define the information exchanged at the rLM, rMTP3user (in SUA: rSCCPuser) and LM communication interfaces. Since the rLM and rMTP3user/rSCCPuser ports are realized via an XML message exchange both have been defined as message-based ports. The types have been derived from the interface related specifications Q.701 and SCCP. The LM interface (to manage SCTP shutdown, reset etc.) has also been defined as a message-based port due to the simple kind of information to be exchanged.

2.4 Behavior definition

One of the first questions w.r.t. the test behavior is about the introduction of parallel test components (PTCs) in addition to the main test component (MTC). According to the test plan for our SIGTRAN tests no need for PTCs have been identified since most tests base on a client-server configuration. There are a few ‘Relay’ configurations, but their test logic is simple and do not force usage of PTCs that have to be managed and coordinated, since this enlarge the length of the TTCN-3 specification.

In the development of the test behavior for SIGTRAN conformance test we followed the clear separation of test pre- and postambles from the core test body. The preambles cover the mapping of the test component ports and SUT interfaces as well as required steps to reach a particular initial protocol state. The actual set of preamble functions depends on the list of test configurations retrieved from the test plan, i.e. the number of involved processes and SCTP streams in relation to the SG / AS under test. Default functions (TTCN-3 altsteps) are restricted to the handling of unexpected receipt of protocol messages or management primitives and lead to the submission of ASPdown messages and the shutdown of the used SCTP associations.

Due to the elementary structure of the test cases we introduced only some auxiliary functions, e.g. for the optional (de)registration of ASPs at a SG, or a message traffic balancing procedure in case of a load sharing between multiple ASPs. Only two external functions have been introduced. They are used to signal special test case runs to the SA in order to consider e.g. modified test suite parameters.

In the following a typical sample test case from the M3UA test suite is presented. According to the test configuration the map and unmap statements (hidden in auxiliary functions) and pre- and postamble are called at the beginning and end of the scenario. The test body itself consists of an outgoing ASPactive message and the receipt of an acknowledgement and state change notification. One timer to prevent deadlocks has been introduced. There are some extra logging statements in the test body to simplify the reading of the test logs in case of test runs ending with a fail verdict. Predefined text strings are assigned to the auxiliary charstring variable "auxlog" that will also be quoted in case of the occurrence of unexpected behavior.

```
testcase M3UA_SGP_ASPTM_V_003() runs on M3UA_Tester system M3UA_System {
  // map LM/M3UA[0]
  handle_map_ConfigA_lstream(self, system, false, false);

  var default nodeadlock := activate (default1());

  initASPinactive_reg();
  log (logStartTB);

  M3UA[0].send (aspActive_s_1(rcP));
  auxlog := TReason_MissingASPActiveAck_rcP;
  Td.start;
  alt
  {
    [] M3UA[0].receive (aspActiveAck_r_1(tmtDefault, ?));
  };
  auxlog := TReason_MissingNotify_Active;
  alt
  {
    [] M3UA[0].receive (
      notify_r_1(AS_STATE_CHANGE_STYPE, AS_ACTIVE_SINFO)
      {Td.stop}
    );
  };

  log (logEndTB);
  PostambleA();

  // unmap LM/M3UA[0]
  handle_unmap_ConfigA_lstream(self, system, false, false);

  setverdict (pass);
}
```

Informal TTCN-3 "with" statements have been added to every test case to support any test suite documentation features. In the following we give the test description from the previous sample test case:

```
with {
  extension "Description: Ensure that the IUT, upon reception of an ASP Active message,
  responds with an ASP Active ACK and sends a NOTIFY message indicating the AS state
  change to AS-ACTIVE. ";
  extension "Preconditions: TestConfiguration A - Successfully established SCTP
  association between the SGP and the ASP. One ASP configured in AS. ASP marked as ASP-
  INACTIVE at the SGP. ";
  extension "Reference: Section 3.8.2 RFC 3332, Section 4.3.1 RFC 3332, Section 4.3.2 RFC
  3332, Section 4.3.4.3 RFC 3332. ";
}
```

Although the TTCN-3 group statements do not have any semantics (e.g. naming scopes) they have been introduced in the ATS to reflect the TSS and to allow an easy access and navigation to test cases since an outline feature of TTCN-3 editors may list the tests according to the group statements.

2.5 PICS/PIXIT parameter

In TTCN-3 both Protocol Implementation Conformance Statement (PICS) and Protocol Implementation eXtra Information for Testing (PIXIT) parameters are implemented as module parameters. The M3UA test suite has 60 module parameters and in SUA there are even some more due to the additional SCCP functionality. We have divided the module parameters in two groups depending on their relationship to the test execution environment (test system IP address). This approach allows the selection of a set of predefined module parameters (defaults) by switching to a particular setting_myenv.tcn3 module while the rest of the ATS parameters are still part of the common data module.

3. ADAPTER IMPLEMENTATION AND TOOLING

The execution of every TTCN-3 ATS against a SUT requires the use of a TTCN-3 compiler and runtime environment but also the implementation of a SA according to the standardized TTCN-3 runtime interface [TRI] definition, i.e. particular methods like triMap, triSend etc. have to be implemented. We applied a Java-based TTCN-3 test tool environment [TTwb].

In case of SIGTRAN the communication to the SUT requires a SCTP layer. We applied the public available SCTP library (SCTPlib) [SLIB] to setup and manage SCTP associations. Since the SCTPlib is developed with C the resulting library has to be connected with the Java-based SA. Beside JNI we had the possibility to use a simple UDP-based datagram mechanism between the SA and a wrapper process to make use of SCTPlib (developments like the TT-datagram [TTme] are also implementations of a distributed TTCN-3 Runtime Interface [TRI]). Consequently in this situation e.g. a setup of a SCTP association is implemented by delivering related SCTP parameters like number of streams in an UDP data packet.

An additional major issue for the adaptation of the SUT and the test system is the implementation of a coder/decoder. It is possible to semi-automate this task by generating the codec. In this case extra information required by the CD on particular message identifiers need to be provided via an extended type information using again the TTCN-3 “with” statements. In following a sample is given for a M3UA DATA message:

```
type record DATA {
    CommonHeader commonHeader,
    DATAParameters p
}
with { encode "isPDU=true; present=bytes(2,1,0x01) && bytes(3,1,0x01);";};
```

The operators check the presents of octets within a byte stream.

4. VALIDATION AND TEST CAMPAIGNS

As in every software development project a quality assurance procedure has to be applied before the product could be delivered to customers. The application of static semantic checks, TTCN-3 compiler runs and even the compilers of the target programming language do help to identify errors to some extent. But additional checks on the dynamic behavior are essential for the test suite quality. If there is no complete SA available it is possible to test data evaluation procedures on expected incoming data by “input” hand-written data templates. They may be applied to the specified regular expressions and patterns. Furthermore SUT behavior could be modeled by other test components that communicate with the test components via test system coordination points instead of external ports.

The CD implementation could be tested through a so-called “Codec validation framework” that makes use of data templates to be sent. Such frameworks simply encode/decode the data in a loop. We recommend evaluating the completeness of all outgoing data by an automatic collection of all data messages to be sent. This is possible if they are identified e.g. by some common suffix “_s”.

With the availability of the SA there is also the possibility of writing so-called mirror tests running on another TTCN-3 test system and communicating via real SCTP associations. This should be done for selected test configurations in order to identify some real-time (e.g. racing) or test coordination problems. Of course this is an ideal possibility to validate the SA and CD, too.

Last but not least test runs against any SUT help to discover additional mistakes. Unfortunately this is not always possible since e.g. the SUT implementation has not been completed or is not available in time for the test developers.

Another additional challenge to validate a test suite is the participation at interoperability tests at some plug test events, e.g. as organized by ETSI [PT], since in some way a conformance test system can be understand as its own SUT implementation.

5. PROJECT EXPERIENCES

5.1 Findings in the RFCs

During the duration of the project several issues in the RFCs have been identified and it shows that an ATS development forces an in-depth study of the base standard and of course helps to improve it. Some of them were reported to the IETF SIGTRAN mailing list [SML]. Clarifications have been received quite quick w.r.t. message parameter structures and ordering and the association of messages to SCTP streams. Questions on message responses in error situations have been more complicated, e.g. a question on a particular error code in case of invalid routing keys didn't get a final answer.

5.2 Project management

The production of software tests is time critical since tests often need to be available before the SUT implementation could be released. In most cases, a formal SUT model is missing and the only approach to get the tests ASAP is the traditional hand-written development process. We had just a couple of weeks for the delivery of our test suites. But the strict application of the test suite development process discussed in the former sections allows a straight forward production of the test products.

Parallel independent work is possible of course between the ATS and the SA implementation work. On the ATS side we had split the tasks w.r.t. test component roles (i.e. ASP or SGP) since the protocol is not symmetric and a lot of procedures do apply to just one protocol side. If an additional developer will join the team later it is possible to leave the data templates for incoming data open in the first draft (as far as the data is not needed in subsequent test steps) and to consider the details on such conformance constraints later.

5.3 Comparison of the M3UA and SUA tests

Only some months after the M3UA test suite have been realized the SUA ATS project has been started. Thus our knowledge on the strategies and solutions was fresh. The maturity of the M3UA test concepts allowed an adaptation to SUA easily. Major differences were due to a group of so-called "common" tests on connection oriented messages to be applied for both sides, the ASP and SGP. The quality of the available test plan was lower than in the M3UA case, e.g. due to wrong test configurations that had to be corrected before implemented in TTCN-3. From the test developers viewpoint the SUA tests could be understood as an extension to the M3UA tests. TTCN-3 types and templates for M3UA or the SCTP related parts of the SA (due to the same underlying layer) have been partially reused, i.e. the work concentrated on an abstract level on the preparation of the test business logics in SUA based on an existing TTCN-3 framework.

6. CONCLUSION

In this project the development steps of industrial TTCN-3 test suites have been presented. The resulting products [TT] have been successfully used in several test campaigns against different SIGTRAN stacks. In comparison to the [M3TP] standard the number of M3UA tests in [TT] is even larger. This is due to the request of additional test cases proposed by customers and shows that due to the abstraction level of TTCN-3 end-users can easily use and enhance a TTCN-3 framework w.r.t. their particular needs.

From the management viewpoint it appears that manual-written ATS development in TTCN-3 is reasonable and subject for an assembly line production process. Further experiences retrieved from similar projects should be collected in some knowledge database like the TTCN-3 forum [TT3].

Acknowledgments

We like to thank our colleagues D. Borowski, G. Din, S. Pietsch and T. Vassiliou who realized other large TTCN-3 test suites for their advices and support.

References

- [CN] ETSI European Standard (ES) 201 873-1 V2.2.1 (2003-02 Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language
- [CTMF] ISO 9646 / ITU-T: Conformance Testing Methodology Framework, multipart standard.
- [D04] T. Deiss: TTCN-3 for Large Systems. In: Systems Validation workshop. Paris 2004.
- [D98] C. Deroches et al.: Test Suite Migration Experiences for B-ISUP and B-MTP. In: IWPTS, Tomsk 1998
- [GFT] ETSI European Standard (ES) and Draft European Standard (DES) 201 873 (2002): The Testing and Test Control Notation version 3 (TTCN-3). Part 3 (ES) Graphical Presentation Format for TTCN-3.
- [IPv6] ETSI TS 102 351: TTCN-3 IPv6 Test Specification Toolkit, ETSI 2004.
- [M3TP] ETSI TS 102 381 V1.1.1, Methods for Testing and Specifications (MTS); SS7 Message Transfer Part 3 - User Adaptation Layer; (IETF RFC 3332); Test Suite Structure and Test Purposes (TSS&TP). December 2004.
- [M3UA] RFC 3332, Signaling System 7 (SS7) Message Transfer Part 3 (MTP3) - User Adaptation Layer (M3UA). September 2002.
- [PT] ETSI SIGTRAN plugtests: <http://www.etsi.org/plugtests/History/2005SIGTRAN.htm>.
- [SIGT] RFC271 IETF: SIGTRAN architecture
- [SIP] ESTI TS 102 027-2: TIPHON Technology Compliance Specification, IETF SIP 3261, ATS and partial PIXIT. ETSI 2003.
- [SCTP] RFC 2960, "Stream Control Transmission Protocol", SCTP: Stream Control Transmission Protocol. October 2000
- [SLIB] SCTPlib: <http://www.sctp.de/sctp-download.html>
- [SML] Sigtran mailing list <https://www1.ietf.org/mailman/listinfo/sigtran>
- [SUA] RFC 3868, Signaling Control Part User Adaptation Layer (SUA). October 2004.
- [TRI] ETSI European Standard (ES) 201 873-1 V2.2.1 (2003-02 Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI)
- [TT] TTsuite user manuals: <http://www.testingtech.de/download/TTsuite-SUAUsersGuide.zip> and [/TTsuite-M3UAUsersGuide.zip](http://www.testingtech.de/download/TTsuite-M3UAUsersGuide.zip)
- [TT3] ETSI TTCN-3 forum: <http://www.ttcn-3.org>
- [TTme] ITEA Project TT-Medal: Test Infrastructure and Architecture. Project deliverable D1.1.1. http://www.tt-medal.org/results/assetbox_process.html. December 2004.
- [TTwb] Testing Technologies TTCN-3 toolset TTworkbench. http://www.testingtech.de/products/ttwb_intro.php
- [W01] A. Wiles et al.: Experiences of using TTCN-3 for Testing SIP and OSP. ETSI, Sophia-Antipolis 2001.
- [W05] C. Willcock et al.: An Introduction to TTCN-3. Wiley, 2005.