ETSI TR 102 976 V1.2.1 (2009-12)

Technical Report

Methods for Testing and Specifications (MTS);
Mobile Reference tests for TTCN-3 tools

Reference

RTR/MTS-00104[2]-MobRefTests

Keywords

ATS, mobile, testing, TTCN

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2009.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™, **TIPHON**™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
LTE™ is a Trade Mark of ETSI currently being registered
for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights 4
Foreword....4
Introduction 4
1 Scope 5
2 References 5
2.1 Normative references 5
2.2 Informative references....5
3 Definitions and abbreviations....6
3.1 Definitions....6
3.2 Abbreviations 6
4 Concepts of a TTCN-3 tool assurance test system....7
4.1 General concepts of mirroring....7
4.1.1 Test data....8
4.1.2 Test component configuration 8
4.1.3 Test behaviour 10
4.1.4 Test verdicts....10
4.2 Test system architectures....10
4.2.1 Intra TTCN-3 Executable approach....11
4.2.1.1 Adaptation layer 13
4.2.2 Inter TTCN-3 Executable approach....14
4.2.2.1 Adaptation layer 16
4.2.3 Comparison of the two approaches....17
5 Abstract Test Suites (ATS)....19
5.1 Test Suite Structure (TSS)....19
5.1.1 Introduction....19
5.1.2 General assumptions 19
5.1.3 System Under Test....19
5.1.4 Test Suite Structure....19
5.2 PICS/PIXIT 20
5.3 TTCN-3 coverage....20
5.4 The ATS in TTCN-3 core (text) format 20
Annex A: Definition of the Inter TTCN-3 Executable interface 21
A.1 CoDec....21
A.2 Test Adaptation 21
Annex B: Feature coverage details 23
Annex C: Using checkpoints to trace the executed paths in the reference ATS 26
History 27

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

Introduction

The present document consists of two distinct parts. In the first part, concepts of a quality assurance system for Testing and Test Control Notation version 3 (TTCN-3) tools are presented. This includes the discussion of general concepts for developing a TTCN-3 mirror test suite for a TTCN-3 reference test suite. In the second part, Abstract Test Suites (ATSs) for a reference TTCN-3 test suite and a mirror TTCN-3 test suite to enable testing of TTCN-3 tools are described.

1 Scope

The present document describes a test system and underlying reference and mirror tests for assuring the correctness of tools for executing tests that are specified using TTCN-3. The scope is the TTCN-3 Executable (TE) as standardized in ES 201 873-1 [i.1] and ES 201 873-4 [i.2]. For the quality assurance of tools that realise the TE, TTCN-3 test specification that are based on reference tests for the 3G mobile telecommunication system are used. This reference test suite and a corresponding mirror test suite are executed by TTCN-3 tools and run against each other to gain confidence in the correctness of the involved TTCN-3 tools.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
    - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
    - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

Not applicable.

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

[i.1] ETSI ES 201 873-1 (V3.3.2): "Methods for Testing and Specification (MTS);The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".

[i.2] ETSI ES 201 873-4 (V3.3.1): "Methods for Testing and Specification (MTS);The Testing and Test Control Notation version 3; Part 4: TTCN-3 Operational Semantics".

[i.3] ETSI ES 201 873-5 (V3.3.1): "Methods for Testing and Specification (MTS);The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI)".

[i.4] ETSI ES 201 873-6 (V3.3.1): "Methods for Testing and Specification (MTS);The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI)".

[i.5] ETSI ES 201 873-7 (V3.3.2): "Methods for Testing and Specification (MTS);The Testing and Test Control Notation version 3; Part 7: Using ASN.1 with TTCN-3".

[i.6] ISO/IEC 9646-1 (1994): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".

[i.7] ISO/IEC 9646-3 (1998): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 3: The Tree and Tabular Notation (TTCN)".

[i.8] ISO/IEC 7498-1 (1994): "Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model".

[i.9] ISO/IEC 10731 (1994): "Information technology - Open System Interconnection - Basic Reference Model: Conventions for the Definition of OSI Services". (Also published as ITU-T X.210 (11/93)).

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 201 873-1 [i.1], ES 201 873-5 [i.3], ES 201 873-6 [i.4], ISO/IEC 9646-1 [i.6], ISO/IEC 9646-3 [i.7] and the following apply:

Main Test Component (MTC): See ISO/IEC 9646-3 [i.7].

mirror ATS: ATS that serves as an SUT replacement for the reference ATS by mirroring the reference ATS

Parallel Test Component (PTC): See ISO/IEC 9646-3 [i.7].

reference ATS: existing ATS that is usually (but not necessarily) derived from some base standard

System Under Test (SUT): See ISO/IEC 9646-1 [i.6].

test system: See ISO/IEC 9646-1 [i.6].

TTCN-3 Control Interface (TCI): See ES 201 873-6 [i.4].

TTCN-3 Executable (TE): See ES 201 873-5 [i.3].

TTCN-3 Runtime Interface (TRI): See ES 201 873-5 [i.3].

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3GPP	3G Partnership Project
AGPS	Assisted GPS
ASN.1	Abstract Syntax Notation 1
ASP	Abstract Service Primitive
ATS	Abstract Test Suite
CD	Coding/Decoding
CH	Component Handling
ETS	Executable Test Suite
GERAN	GSM Edge Radio Access Network
MTC	Main Test Component
OSI	Open Systems Interconnection
PA	Platform Adapter
PDU	Protocol Data Unit
PER	Packed Encoding Rules of ASN.1
PTC	Parallel Test Component
RAB	Radio Access Bearer
RLC	Radio Link Control
RRC	Radio Resource Control
SA	SUT Adaptor

SUT	System Under Test
TCI	TTCN-3 Control Interface
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TE	TTCN-3 Executable
TL	Test Logging
TM	Test Management
TRI	TTCN-3 Runtime Interface
TSI	Test System Interface
TTCN-2	Tree and Tabular Combined Notation version 2
TTCN-3	Testing and Test Control Notation version 3
UDP	Unreliable Datagram Protocol
XML	Extensible Markup Language

4 Concepts of a TTCN-3 tool assurance test system

The aim of the TTCN-3 tool assurance test system and the underlying reference tests described in the present document is to assess the standard compliance of TTCN-3 tools in order to improve the exchangeability of TTCN-3 test specifications between TTCN-3 tools from different vendors. In the current approach, the focus is the TTCN-3 Executable (TE) including interpretation or compilation and execution of TTCN-3, i.e. ES 201 873-1 [i.1] and ES 201 873-4 [i.2] are covered. The additional entities of the general structure of a TTCN-3 test system (such as the TTCN-3 Control Interface (TCI) and the TTCN-3 Runtime Interface (TRI)) are not in the scope of the present document.

The present document treats TTCN-3 tool assurance in relation to the 3G Partnership Project (3GPP) mobile telecommunication system; hence, existing Tree and Tabular Combined Notation (TTCN-2) test cases that have been standardized by 3GPP have been selected and specified as a reference Abstract Test Suite (ATS) in TTCN-3. This includes also Abstract Syntax Notation 1 (ASN.1) modules that are used as described in ES 201 873-7 [i.5]. This reference test suite can be used to test the standard compliance of TTCN-3 tools. For testing the compliance with respect to syntax and static semantics as defined in ES 201 873-1 [i.1] (and with respect to ASN.1 also ES 201 873-7 [i.5]), it is sufficient to let the assessed TTCN-3 tool process the TTCN-3 modules. However, to enable testing of compliance of a TTCN-3 tool with respect to the operational semantics as described in ES 201 873-4 [i.2], the reference ATS needs to be executed.

For executing the reference ATS, a TTCN-3 test system needs to be realised and another system to communicate with needs to exist. Usually, there is a System Under Test (SUT) against which the TTCN-3 test system is executed. To remove the dependency from a real SUT, the approach described in the present document realises the SUT also by a TTCN-3 test system. As a result, there are two ATSs: the reference ATS and a mirror ATS that mirrors the reference ATS. By running the TTCN-3 test system that executes the reference ATS and TTCN-3 test system that executes the mirror ATS against each other, a back-to-back test comparable to an interoperability test at PLUGTESTS events is performed. Even though this is a typical interoperability test setup, the setup results from a conformance test where the SUT is a TTCN-3 test system that is tested by another TTCN-3 test system.

According to the described approach, a mirror ATS needs to be created in addition to the reference ATS. Even though a mirror ATS needs to be individually designed for each reference ATS, some general concepts of mirroring a test suite are described in the next clause.

4.1 General concepts of mirroring

As a replacement for the "real" SUT that is assumed by the reference ATS, the reference ATS is "mirrored" yielding a mirror ATS. The creation of an ATS as a mirror for an SUT is a standard approach in test development. However, for the creation of a mirror ATS for another ATS some special considerations apply. These general concepts of mirroring an ATS are discussed in the following.

The process of mirroring a TTCN-3 ATS consists of mirroring relevant entities, e.g. test case and function definitions, port and component definitions, template definitions, of the reference ATS. To make the relation between mirror definition and reference definition clearer, in the following, the suffix `_mirror` is added to identifiers of definitions in the mirror ATS that are a direct mirror of a definition in the reference ATS. This name convention introduces a traceability and increases furthermore the maintenance of e.g. the mirror ATS if the reference ATS is changed. In addition to this naming convention, the approach described in the present document suggests to use a similar TTCN-3 module structure in the mirror ATS as in the reference ATS: those modules of the reference ATS that can be re-used without change are imported by the mirror ATS; however, when it is necessary to change a module, it can be copied, renamed using the suffix `_mirror`, and changed as necessary thus resembling the original structure of the reference ATS.

The mirroring of definitions refers in particular to those concepts that are related to test data, test component configuration, and test behaviour. Furthermore, when running a mirror ATS against a reference ATS, the meaning of test verdicts needs to be discussed.

4.1.1 Test data

In the telecommunication area, the layered OSI Basic Reference Model [i.8] is widely used. This includes the usage of a service by means of Abstract Service Primitives (ASP) that result in the exchange of Protocol Data Units (PDU) between peer protocol entities [i.9]. An ASP conveys information from a service-user to a service-provider and vice versa. Usually, the service-provider applies some sort of mirroring of the ASP type between the origin service-user and the target service-user, i.e. a request ASP is turned into an indication ASP and a response ASP into a confirmation ASP.

However, for the back-to-back test setup described in the present document, a simplified communication model is used and hence, there is no service provider between the two ATSs that mirrors the ASPs - instead the mirror ATS will receive the message that was sent by the reference ATS without any change and vice versa. As a result, the type definitions that are used in the reference ATS for specifying ASPs need not to be mirrored in the mirror ATS, but can be re-used without change in the mirror ATS. To some extent this applies as well for TTCN-3 template definitions; however, as part of the mirroring, templates used for sending are now involved in receiving and vice versa. Hence, template definitions need to be changed with respect to the usage of wildcards and other matching mechanisms. Nevertheless, template definitions from the reference ATS can be re-used by using them in the mirror ATS as base template and deriving new templates from them using the TTCN-3 **`modifies`** keyword: a send template from the reference ATS can be adapted to a receive template by allowing wildcards where needed, a receive template from the reference ATS can be turned into a send template by overwriting wildcards with concrete values.

4.1.2 Test component configuration

The component type definitions of the reference ATS refer to the ports through which a test component is connected to other test components or mapped the test system interface (TSI). For supporting the back-to-back execution of reference ATS and mirror ATS against each other, those test component ports of the reference ATS that are mapped to the TSI need to be connected to corresponding mirror ports of components of the mirror ATS. This is shown in figure 1: Ports p1, p4 and p7 of the reference ATS are mapped to the TSI (Part (a) of figure 1). In the resulting back-to-back test setup (Part (b) of figure 1), these ports are connected to the respective ports p1_mirror, p4_mirror and p7_mirror of the mirror ATS. To allow this connection of ports, the port type definitions of the mirror ports need be mirrors of the reference port definitions, i.e. the port type definitions need to be copied and the therein specified directions of the message and/or procedure based communication need to be inverted (i.e. the direction specification **`in`** is changed into **`out`** and vice versa). However, as described in clause 4.1.1, the types that are conveyed via a port can remain unchanged.

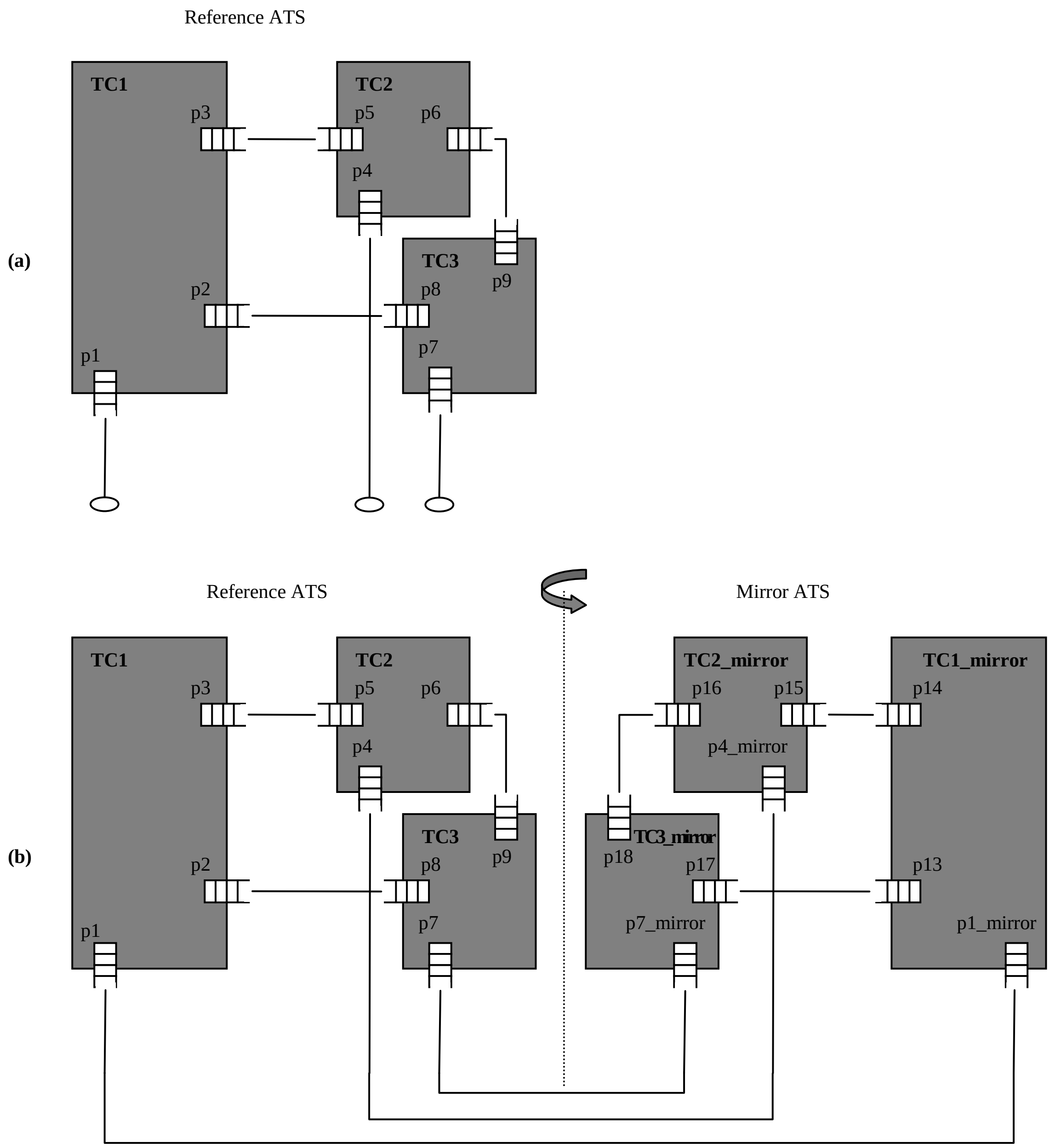

Figure 1: An example reference ATS (a) and the resulting back-to-back test setup (b)

As a design decision, it needs to be determined whether only a single test component or several parallel test components are used as test component configuration for the mirror ATS. To aid traceability and increase maintainability between reference and mirror ATS, the approach described in the present document uses the same test component configuration in the mirror ATS as in the reference ATS: as shown in figure 1 (b), there are components TC1_mirror, TC2_mirror and TC3_mirror in the mirror ATS that resemble the components TC1, TC2 and TC3 of the reference ATS. Whether the ports of the reference ATS that are used for the communication between reference ATS components need also to be mirrored or at least copied depends on the internal logic of the reference ATS and mirror ATS - thus, this cannot be decided in general. The same considerations apply for the decision whether to copy any other elements of a reference component into a mirror component, i.e. component constants, variables, and timers.

4.1.3 Test behaviour

While the definitions of data in the reference ATS can often be re-used in a mirror ATS, the test behaviour of the mirror ATS needs to be re-written, as it represents the mirrored behaviour of the reference ATS, e.g. a **`send`** operation needs to turned into a **`receive`** operation and vice versa. When mirroring test behaviour, one mirror test case that runs against a reference test case executes exactly one path of the reference test case. However, usually the reference test case contains branching, e.g. due to an **`alt`** construct, where one branch of the reference ATS leads e.g. to setting the test verdict **`pass`**, and another branch to test verdict **`fail`**. It depends on the coverage criteria of the mirror ATS, which and how many of these branches are to be covered. Hence, this results in general in multiple mirror test cases for one reference test case, because each mirror test case aims at executing a different path of the reference test case.

During the development of a mirror test case, an assumption is made which path is followed in the reference ATS. Hence, usually the mirror test case may assume to make exactly one certain observation, e.g. by using a **`receive`** operation without any further explicit alternatives (however it is reasonable to activate a default **`altstep`** for handling any unexpected events).

Even though the mirror test behaviour needs to be re-written, it is reasonable to keep the same structure as in the reference ATS: e.g. the same division of test behaviour into functions for pre- and postambles or test steps can be kept. However, it might e.g. not be reasonable to retain **`altstep`** definitions from the reference ATS, because the branching that is introduced in reference test cases by **`alt`** statements and **`altstep`** definitions, results - as described above - in multiple mirror test cases (depending on the coverage criteria).

During the execution of the mirror ATS against the reference ATS, it needs to be assured that the execution of the different test cases is synchronized, i.e. that matching reference test cases and mirror test cases run against each other. If several branches of the same reference test case are tested by several mirror test cases, this requires that the same reference test case is executed several times. Special care needs to be taken that both, the reference test case and the mirror test case, finish their postambles at the same time, to prevent that one test suite is still executing a postamble of a test case and the other test suite has already started to execute the preamble of its subsequent test case.

Some part of the reference test behaviour may be defined by external functions. If the reference test suite is not executed against a real SUT, but in the fixed context of a back-to-back TTCN-3 tool assurance test setup, there may be the possibility to replace external functions by internal TTCN-3 functions, thus reducing the implementation efforts: it may be the case that it is either possible to predict that only a fixed value needs to be returned by an external function and thus an internal TTCN-3 function that returns a hard-coded value is sufficient or it is possible that a certain side-effect of an external function is not required (e.g. for configuring some external test hardware) and thus does not need to be implemented in the context of reference test suite and mirror test suite.

4.1.4 Test verdicts

In the back-to-back TTCN-3 tool assurance test setup, two ATSs are executed against each other: the reference ATS that has been developed for the conformance test of a real SUT and the mirror ATS that has been developed for the conformance test of a TTCN-3 test system that executes the reference ATS (and implicitly the a TTCN-3 test system that executes the mirror ATS is also tested.) Both ATSs are setting TTCN-3 test verdicts that need to be interpreted. While the mirror ATS has been designed to set a **`pass`** verdict if it regards the involved TTCN-3 tools as conforming and **`fail`** if not, this is not the case for the reference ATS as it has been designed to test a real SUT. As part of the TTCN-3 tool assurance test it may - depending on the coverage criteria - thus be the case that the mirror ATS intentionally drives the reference ATS into a branch that sets e.g. a **`fail`** or **`inconc`** verdict. However, these verdicts do not indicate that the TTCN-3 tool assurance test has failed or is inconclusive. (It may even be the case that the reference ATS did set a **`pass`** verdict, however according to the test purpose of the mirror ATS, a **`fail`** verdict was expected). Hence, for the interpretation of the verdicts set by the reference ATS, a detailed analysis of each test run is required. (An advanced approach is discussed in annex C.)

4.2 Test system architectures

In the following, two different architectural approaches for mirroring a TTCN-3 test system that are suitable for a back-to-back test are presented. This includes also the discussion of how to actually run the reference and the mirror ATS against each other, by turning them into an Executable Test Suite (ETS): in addition to a TTCN-3 tool that realises the TE from the ATS, this requires a System Adapter (SA), Coding/Decoding (CD) and - in case of external functions - also a Platform Adapter (PA). All other entities of a TTCN-3 test system (as well as the PA part that provides access to the platform timer) are assumed to be provided by the TTCN-3 tools.

4.2.1 Intra TTCN-3 Executable approach

The Intra TTCN-3 Executable approach is based on multiple TTCN-3 components within a single TE, forming a self-contained and stand-alone test system that can be executed without any external entity (i.e. no entity that corresponds to an external SUT is required). This approach allows a back-to-back interoperability test by utilizing a single ETS that runs several TTCN-3 components against each other within the same TTCN-3 test system. The principle structure is illustrated in figure 2.

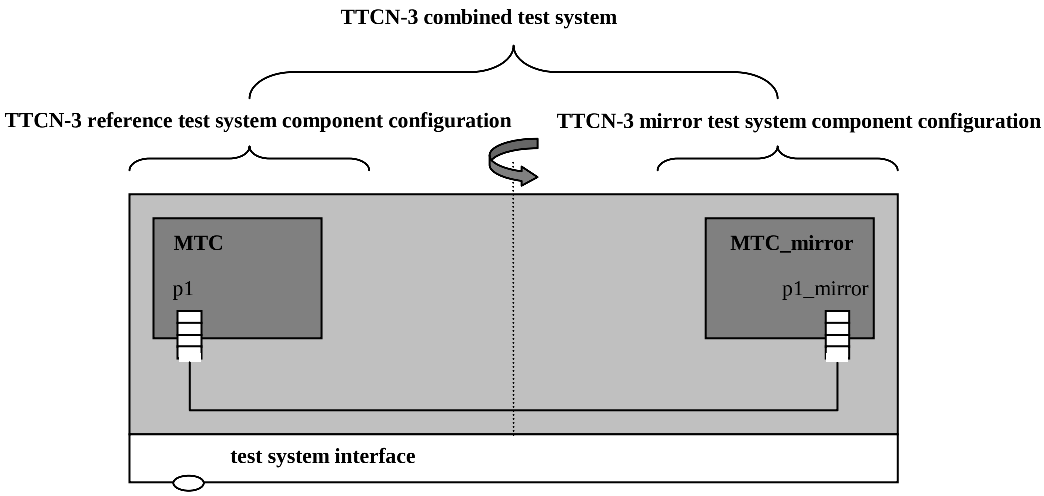

Figure 2: Illustration of Intra TTCN-3 Executable Approach

In the TTCN-3 component configuration shown in figure 2, roles can be assigned to the involved TTCN-3 components as follows:

- TTCN-3 reference test system component configuration:
  This role refers to the TTCN-3 component architecture of the reference ATS.
- TTCN-3 mirror test system component configuration:
  This role refers to the TTCN-3 component architecture of the mirror ATS.
- TTCN-3 combined test system:
  This role refers to the TTCN-3 test system that is formed out of the reference and the mirror ATS being executed within a single TE.

NOTE: In the present document, only message-based communication principles will be regarded. Nevertheless, the Intra TTCN-3 Executable approach can be also applied to procedure-based communication principles.

In cases where the component configuration used in the reference ATS consists of an MTC and one or several PTCs, a possible approach is to mirror this component configuration in the mirror ATS as well. An example is depicted in figure 3: part (a) shows a test component architecture that is utilizing multiple components as MTC and PTCs. The components are assumed to be connected, as realised by the TTCN-3 **`connect`** statement. Further on, the components have connections towards the TSI, as realised by the **`map`** statement.

figure 3(b) shows the combined test system that is derived from the test component architecture shown in figure 3(a): The combined test system includes the component structure of the reference ATS and the mirror ATS. The mapping of the ports of the reference ATS towards the TSI, such as the mapping of p1 to p10, p4 to p11, and p7 to p12 as shown in figure 3(a), needs to be re-directed to the adequate mirror test components, resulting in connection between p1 and p1_mirror, between p4 and p4_mirror, and between p7 and p7_mirror as shown in figure 3(b). As a result, test data can be exchanged accordingly between components of the reference ATS and the mirror ATS.

(a)

(b)

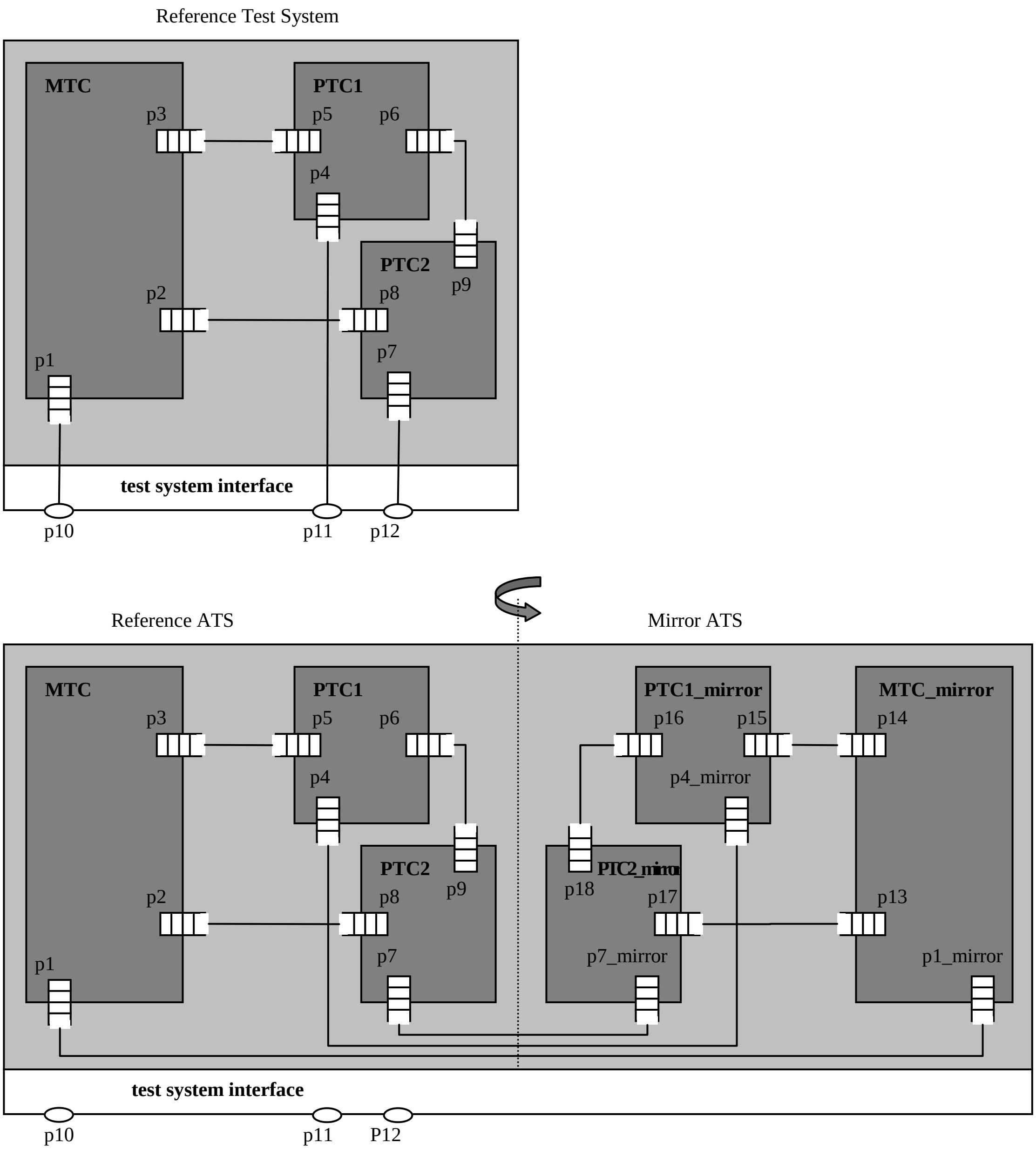

Figure 3: Illustration of multiple component reference test system (a) and derived combined test system (b), including the mirror test multiple component architecture

The structure of the mirror component configuration (figure 3(b)) resembles the reference component configuration as suggested in clause 4.1.2: Each TTCN-3 component of the reference ATS has its own mirror test component.

In order to deploy the reference ATS according to the Intra TTCN-3 Executable approach, the TTCN-3 source code of the reference ATS needs to be modified as follows: mappings between TTCN-3 components and the test system interface are no longer permitted and need to be replaced by proper connections towards adequate TTCN-3 mirror components as depicted in figure 3(b). As a result, there is no mapping towards the test system interface anymore.

When the combined test system is executed, the global test case verdict is calculated based on the combined local test case verdicts of reference ATS and of mirror ATS.

4.2.1.1 Adaptation layer

Assuming that the standardized TRI/TCI architecture is used, the resulting general structure of the Intra TTCN-3 Executable approach looks like given in figure 4. Since only internal communication between test components is involved, the SA entity needs not to implement any communication mechanisms and the CD entity does not decode or encode any external data. As a result, simplified requirements are imposed on the entities that implement the different TRI/TCI interfaces of the involved TTCN-3 tooling.

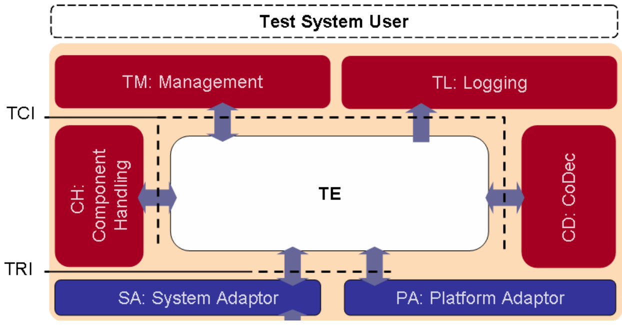

Figure 4: Adaptation of the Intra TTCN-3 Executable approach using TRI and TCI

Following are the requirements to the implementations of the different TRI/TCI interfaces:

a) TL: The Test Logging entity can remain unchanged as provided by each TTCN-3 tool.

b) CH: The Component Handling entity can remain unchanged as provided each TTCN-3 tool.

c) TM: The Test Management entity can remain unchanged as provided by each TTCN-3 tool.

d) CD: A Coding/Decoding entity needs not to be implemented.

NOTE 1: Upcoming TTCN-3 standards might have a new TTCN-3 **`encode`** and **`decode`** functions standardized that allows direct access to TCI-CD.

e) SA: The System Adapter requirements do depend on the utilized TTCN-3 code within the combined system. Per concept the implementation of TRI map/unmap is not required, because TTCN-3 map commands are replaced by TTCN-3 **`connect`** statements.
Further implementation in SA is depending on the usage of TTCN-3 concepts in the combined test system (see below).

f) PA: The Platform Adapter provided by each TTCN-3 tool can be used for timer handling. However, external functions required by either the reference or mirror ATS need to be implemented. It may be possible to replace external functions by simplified internal TTCN-3 functions, thus reducing the implementation efforts for the PA.

The requirements on the SA depends on the utilized TTCN-3 concepts in the TTCN-3 combined test system: Some statement do directly result in a TRI action. In case the combined test system utilizes one of the TTCN-3 statements listed below, adequate support within the SA is required. The following list shows the dependencies between TTCN-3 statements or operations and TRI actions that need to be considered in the Intra TTCN-3 executable approach for the SA requirements.

1) Usage of TTCN-3 **`map`** / **`unmap`**: → triMap / triUnmap:

   i. Map statements in the reference test system are not permitted when running the intra TTCN-3 executable approach and need to be replaced by adequate TTCN-3 **`connect`** operations towards the adequate mirror component.

   ii. Unmap operations in the reference test system are not permitted when running in the intra TTCN-3 executable approach and need to be replaced by adequate TTCN-3 **`disconnect`** operations towards the adequate mirror component.

2) Usage of TTCN-3 **`execute`**: → triExecuteTestCase, triStartTimer, triEndTestCase.

NOTE 2: These TRI operations are utilized for both cases: TCI direct test case execution and control part.

3) Usage of TTCN-3 **`action`**: → triSUTactionInformal.

4) Usage of TTCN-3 **`start`** (timer): → triStartTimer.

NOTE 3: Adequate implementation in PA is also required.

5) Usage of TTCN-3 **`stop`** (timer): → triStopTimer.

NOTE 4: Adequate implementation in PA is also required.

6) Usage of TTCN-3 **`read`** (timer): → triReadTimer.

NOTE 5: Adequate implementation in PA is also required.

7) Usage of TTCN-3 **`running`** (timer): → triTimerRunning.

NOTE 6: Adequate implementation in PA is also required.

8) Usage of TTCN-3 **external function**: → triExternalFunction
The implementation of external functions can be re-used with out problems as long the provided functionality is implemented in the utilized SA. In case of the external function utilizes further operations on e.g. CoDec operations, either CoDec functionality is considered to be provided at run-time or alternatively the external function needs to be stubbed adequate either:

   a) in the external external function; or

   b) the entire external function needs to be stubbed by an adequate regular TTCN-3 function.

NOTE 7: In case none of the above statements are utilized in the TTCN-3 combined test system, a default SA can be utilized that does not implement any functionality.

4.2.2 Inter TTCN-3 Executable approach

The Inter TTCN-3 Executable approach allows a back-to-back interoperability test of possibly two different TTCN-3 tools by executing two separate TTCN-3 test systems against each other:

- One TTCN-3 test system (the referenceTTCN-3 test system) executes the reference ATS.
- The other TTCN-3 test system (the mirror TTCN-3 test system) executes the mirror ATS.

This structure is illustrated in figure 5: Part (a) shows an example test component configuration used in a possible reference ATS: in this test component configuration, the test components have ports that are connected to each other as well as ports that are mapped to the TSI. figure 5(b) shows the two test systems resulting from the Inter TTCN-3 Executable approach. The component configuration of the reference ATS is unchanged. As described in clause 4.1.2, the test component configuration of the mirror ATS resembles the test component configuration of the reference ATS. However, the ports of the TSI and those that are mapped to the TSI are mirrored. The TSI of the reference ATS and the TSI of the mirror ATS are connected to each other via the adaptation layers of both test systems.

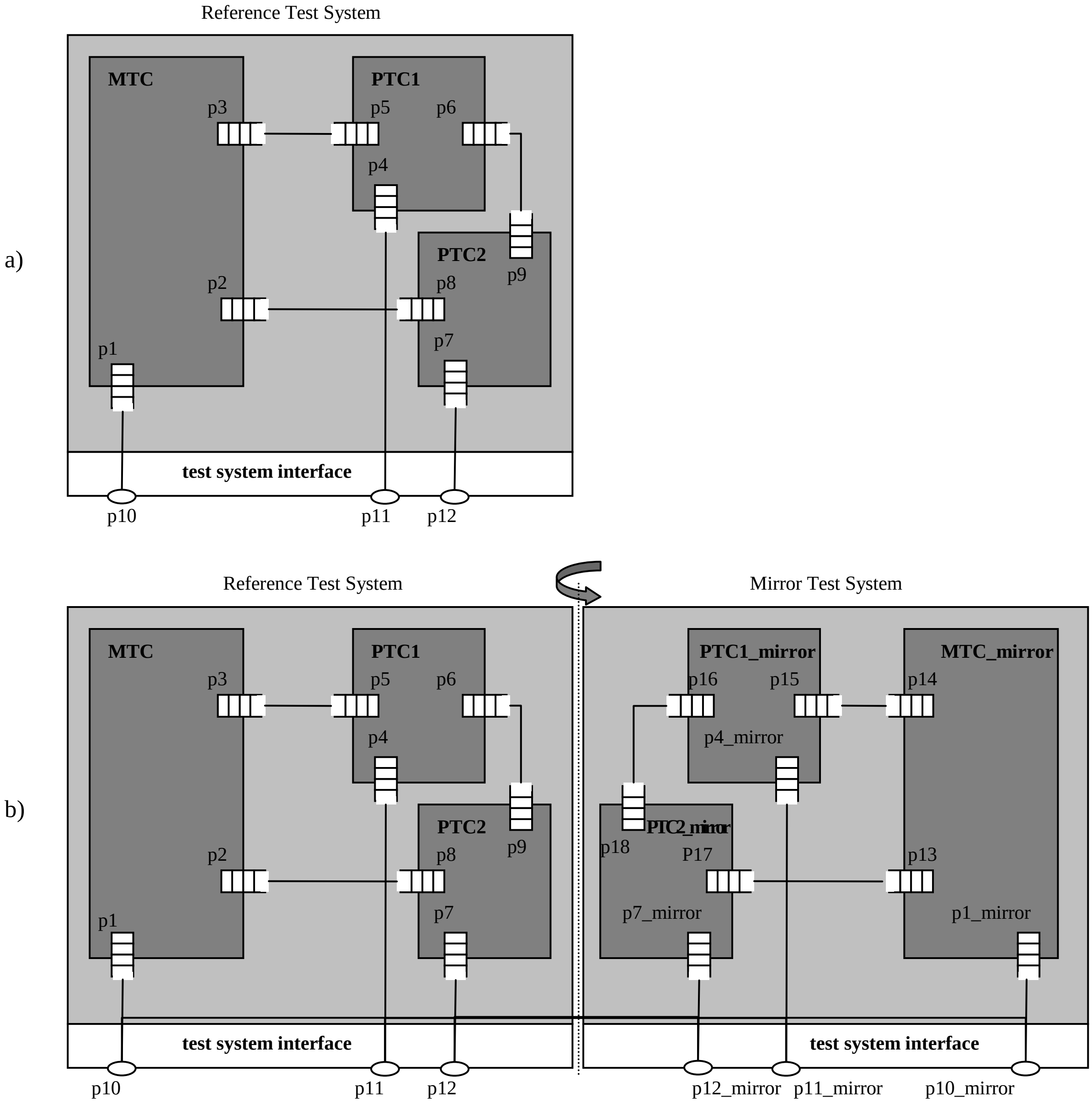

Figure 5: An example reference test system (a) and the resulting architecture according to the Inter TTCN-3 Executable approach (b)

Both TTCN-3 test systems execute independently from each other, e.g. both provide their own test system user interface, both perform their own logging, and both maintain their own global test verdict. Because both TTCN-3 test systems maintain their own global test verdict, it is possible to distinguish easily between the verdict set by the reference test suite and the verdict set by the mirror test suite. As described in clause 4.1.4, the mirror test suite is designed in a way that the verdicts that are set by its test cases provide an assessment of the test system executing the reference test suite. Hence, it is definitely an indicator of a non-conformance in either of the two involved test tools, if the verdict set by the mirror test system is non-pass. However, as described in clause 4.1.4, the interpretation of the verdict set in the reference test system requires a detailed investigation.

For realising the Inter TTCN-3 Executable approach, the component configuration of the reference ATS can remain unchanged, in particular it maps its ports as usual to the TSI. For the mirror ATS, the following requirement is considered to be fulfilled: The TSI of the mirror TTCN-3 test system is a mirror of the TSI of the reference TTCN-3 test system. This means, it possesses the same port structure; however, the ports are mirrored, i.e. the direction of communication is inverted in the mirror TSI. The relevant ports of the mirror ATS components are mapped as usual to the TSI. As described in clause 4.1.2, only the direction, but not the types of the messages are mirrored. This means, that ASPs that are sent or expected by the reference test suite are not mirrored, but received or sent using the same type definitions that are defined in the reference test suite.

To allow the two TEs to communicate with each other, the TSI of the reference TTCN-3 test system and the TSI of the mirror TTCN-3 test system need to be connected to each other. This can be achieved by an adaptation layer that adapts the communication of the two TEs to each other (including the corresponding encoding and decoding of transmitted data).

4.2.2.1 Adaptation layer

Assuming that the standardized TRI/TCI architecture is used, the resulting general structure of the Inter TTCN-3 Executable approach looks like given in figure 6. The encoding and decoding of data is performed by the CD entities, the actual sending and receiving of data values is performed by the SA entities. The interfaces to the test system user are provided by the TM entities.

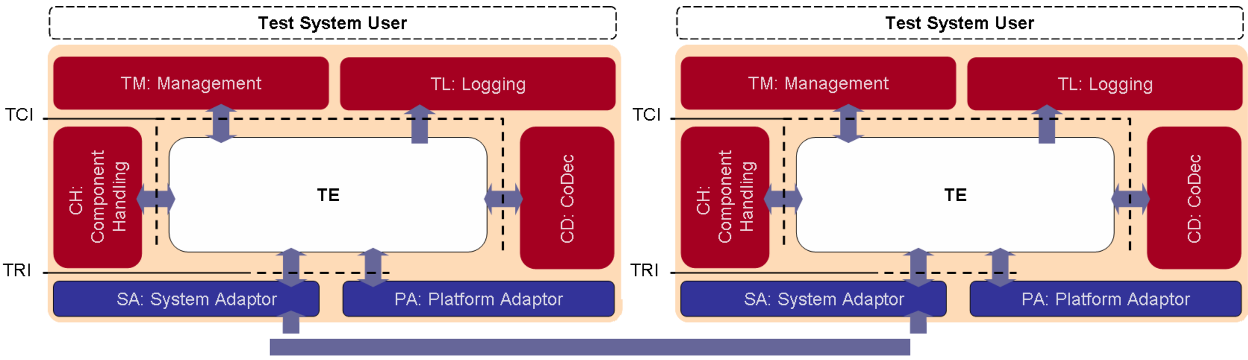

Figure 6: Adaptation of the Inter TTCN-3 Executable approach using TRI and TCI

If the reference test suite that is executed on the reference TTCN-3 test system is a test suite for some base standard (e.g. one from 3GPP), the encoding and decoding of values exchanged by the two TTCN-3 test systems might result from the encoding and decoding rules defined in the base standard (e.g. ASN.1 PER). However, as long as only two TTCN-3 test systems need to communicate with each other and nor real SUT is involved, other encodings that are easier to implement (e.g. XML as suggested in annex A) may be used instead. Furthermore, the transmission of the encoded values needs not to adhere to the communication mechanisms described in a base standard; thus, a less complex communication mechanism may be used for sending and receiving data values (e.g. TCP as suggested in annex A). The SA implementation may be eased by using a distinguished logical communication channel for each connection of a TSI port and its corresponding mirror port.

As long as both TTCN-3 test systems are able to communicate with each other via the adaptation layer, it does not matter whether each TTCN-3 test system is executed on the same platform (hardware and operating system) or on different ones. However, if the two TTCN-3 tools have different platform requirements, they are naturally executed on two different platforms.

Each TTCN-3 test system has its own TM entity that provides the interface to the test user. For a meaningful back-to-back test, a coordinated execution of reference test cases and corresponding mirror test cases is required as discussed in clause 4.1.3. When testing a real SUT, pre- and postambles are used to reset the SUT and to make execution of the test body meaningful and thus allows a continuation with a subsequent test case even if the previous test case failed. However, in the Inter TTCN-3 Executable approach there is no standardized way for one TTCN-3 test system to reset the other test system. Hence, the following three options apply to realise the TM entities of the two test systems:

- Custom implementations for the two TM entities are realised. They execute the reference and mirror test cases in the required order. The two TM entities might be controlled by a master test management entity or communicate as peers with each other to achieve the required synchronization and reset of TTCN-3 test systems.

- Default TM implementations that are provided by the TTCN-3 tools are used together with a TTCN-3 module control part in each test suite. The module control parts realise the coordinated execution of test cases in both, the reference and the mirror test suite, by specifying both times the required order in which test cases are executed. In case that the pre- and postambles of the test cases are not sufficient for synchronization, the TM can be either operated under manual supervision or external functions might be called between consecutive **`execute`** statements. These external function implement communication between the reference and the mirror test system to achieve the required synchronization.

- Default TM implementations that are provided by the TTCN-3 tools are used together with a manually supervised execution of a single pair of reference and mirror test case. While no automated runs of complete test campaigns are possible, this option is easy to realise.

The consequences of the Inter TTCN-3 Executable approach for the entities that implement the different TRI/TCI interfaces for the two involved TTCN-3 tools, are as follows:

a) TL: The Test Logging entity can remain unchanged as provided by each TTCN-3 tool.

b) CH: The Component Handling entity can remain unchanged as provided each TTCN-3 tool.

c) TM: The Test Management entity can remain unchanged as provided by each TTCN-3 tool. However, if synchronization of test case execution is not sufficiently solved by pre- and postambles, it has be considered to add a synchronization between the test management entities of the two involved test tools either by manual start (and termination and reset in case of problems) of test cases or by a tool-based synchronization.

d) CD: A Coding/Decoding entity needs to be implemented that implements a coding scheme, e.g. either according to the base standard for which the reference test suite has been written or a using a simplified encoding that is only applicable as long as no real SUT is involved.

e) SA: A System Adapter that realises a communication mechanism that may be simpler than the one foreseen in the base standard for which the reference test suite has been written can be implemented, e.g. one based on TCP/IP.

f) PA: The Platform Adapter provided by each TTCN-3 tool can be used for timer handling. However, external functions required by either the reference or mirror test suite need to be implemented. It may be possible to replace external functions by simplified internal TTCN-3 functions, thus reducing the implementation efforts for the PA.

4.2.3 Comparison of the two approaches

Both of the two presented approaches have their advantages and disadvantages with respect to efforts required for making the ATSs executable, evaluation of test verdicts, and properties that may be tested. These are shown in table 1 and are discussed in more detail in the respective table notes.

Table 1: Comparison of Intra and Inter approaches, PRO (+) and CONS (-)

Issue	Intra TTCN-3 Executable approach - key arguments	Inter TTCN-3 Executable approach - key arguments
TCI-CD comparison (see note 1)	+: Reduced efforts to make the test suites executable: No requirement for CoDecs	+: Provides principle support for evaluating CoDec properties
SA/PA comparison (see note 1)	+: Reduced efforts to make SA, or at least just a dummy CD and SA - : Utilized external functions need also to be implemented by the SA and are specific to each test suite. Depending on the purpose of the external function this might have further dependencies	+: SA/PA as utilized for final TTCN-3 test system can be utilized and stabilized
TTCN-3 scripting comparison (see note 2)	-: The reference ATS needs to be adapted and enhanced by mirror ATS components	+: The reference ATS does not need to be changed
Comparison on Verdict handling (see note 3)	-: The verdicts set by the reference and mirror components are mixed into a single global verdict	+: The two involved test systems maintain their separate global verdicts
Test depth comparison (see note 4)	-: Higher possibility of not detecting a misinterpretation of the TTCN-3 standard	+: Back to back interoperability tests of possibly two different TTCN-3 tools ensures direct comparison of TTCN-3 artefacts of different TTCN-3 tooling
Overall test system comparison (see note 5)	-: Simplified TTCN-3 test system configuration does result in pure internal data exchange, which might not reflect the real conditions of the TTCN-3 test system	+: All entities of a TTCN-3 test system are crossed. This allows to compare the overall properties of the TTCN-3 tooling, including external entities such as CoDec

NOTE 1: The Intra TTCN-3 Executable approach requires less efforts to make the test suites executable than the Inter TTCN-3 Executable approach. For the latter, the CD and the SA entities of both test systems need to be implemented. In particular if the CD entity cannot be automatically generated, this is tedious work. In contrast, the Intra TTCN-3 Executable approach relies only on component-to-component communication which is already provided by the TTCN-3 tools. Hence, for the Intra TTCN-3 Executable approach no CD and SA or at least just a dummy CD and SA need to be implemented (depending on whether the TTCN-3 tool requires a CD and SA to satisfy its interfaces). Both approaches require a PA to be implemented. Usually, a TTCN-3 tool provides already a default implementation for timer handling; however, external functions need also to be implemented by the PA and are specific to each test suite. As described in clause 4.1.3, it may be possible to replace external functions by internal TTCN-3 functions, thus reducing the implementation efforts.

NOTE 2: While the Inter TTCN-3 Executable approach requires more efforts with respect to providing the required adaptation layer, the reference ATS does not need to be significantly changed in this approach. In contrast, the Intra TTCN-3 Executable approach requires at least a few changes to the reference ATS, e.g. **map** operations need to be replaced by **connect** operations. However the efforts of these changes can be neglected in comparison to the efforts required for making the Intra TTCN-3 Executable approach executable.

NOTE 3: The evaluation of the verdicts set by the reference test suite and by the mirror test suite may be easier in the Inter TTCN-3 Executable approach, because the two involved test systems maintain their separate global verdicts, whereas in the Intra TTCN-3 Executable approach, the verdicts set by the two test suites are mixed into a single global verdict and thus it cannot immediately decided whether a non-pass verdict was set by the mirror test suite that assesses the reference test suite or whether it was set by intentionally executing a non-pass branch of the reference test suite. Thus, it may be considered to remove any verdict setting statements from the reference ATS.

NOTE 4: Due to the fact the Intra TTCN-3 Executable approach involves only one TTCN-3 tool, the possibility of not detecting a misinterpretation of the TTCN-3 standard by that tool may be higher, because by applying the same misinterpretation twice, a fault may be masked. However, this effect may also occur in the Inter TTCN-3 Executable approach. Furthermore, it is important to understand that it is no drawback of the Intra TTCN-3 Executable approach that it does not allow a real back-to-back interoperability test, because a back-to-back operation of different TTCN-3 test tools is not relevant for industrial test practice.

NOTE 5: The fact that the Intra TTCN-3 Executable approach does not require to implement a CD entity has the disadvantage that it is not possible to assess properties of coding and decoding. In contrast, the Inter TTCN-3 Executable approach does allow to test properties of the CD entity. However, the implementation of the CD entity is outside of the scope of a TTCN-3 tool and thus not discussed in the present document.

As a result, the conclusion is that the Intra TTCN-3 Executable approach can be implemented with considerably less efforts, whereas the Inter TTCN-3 Executable approach allows to assess more properties, in particular those of the CD entity.

5 Abstract Test Suites (ATS)

This ATS has been produced using the Testing and Test Control Notation (TTCN) according to ES 201 873-1 [i.1] and ES 201 873-7 [i.5].

5.1 Test Suite Structure (TSS)

5.1.1 Introduction

This ATS consist of two parts, a reference ATS and a mirror ATS. Depending of the chosen test system architecture (see clause 4.2), the reference ATS and the mirror ATS are used as a single combined ATS or as two separate ATSs:

- If the Intra TTCN-3 Executable approach is chosen, the mirror ATS is a combined with reference ATS. The combined ATS –reference ATS and mirror ATS– is run by a single TTCN-3 tool. No test adaptation and no CoDec is needed if the TTCN-3 tool provides a default implementation for them.

- If the Inter TTCN-3 Executable approach is chosen, the mirror ATS is run by one TTCN-3 tool and the reference ATS is run by another TTCN-3 tool (or by a separate instance of the same tool). A test adaptation and CoDec(s) are needed (for a suggestion, see annex A).

5.1.2 General assumptions

The ATS is based on the 3GPP benchmark ATS provided by ETSI MCC TF160. The test cases covering major fields from 3GPP. The ATS has not the claim to cover the TTCN-3 standard itself; further test cases need to be developed to achieve a full coverage of the TTCN-3 standard with all parts.

5.1.3 System Under Test

The notion of system under test is not applicable as it is replaced by the reference and mirror test cases that are executed by an arbitrary TTCN-3 tool that supports ES 201 873-1 [i.1] and ES 201 873-7 [i.5].

5.1.4 Test Suite Structure

The test suite is divided into a reference ATS and a mirror ATS. They provide test cases for the "Main Functionalities" shown in table 2. If the Inter TTCN-3 Executable approach is used, the "Main Functionalities" provide the TTCN-3 reference test cases that need to be run by the reference test system, and the "Main Functionalities Mirror" provide the corresponding TTCN-3 mirror test cases that need to be run by the mirror test system. If the Intra TTCN-3 Executable approach is used, the "Main Functionalities" provide the TTCN-3 reference test cases that do not only contain reference test behaviour, but also take care of connecting reference with mirror components and starting corresponding TTCN-3 functions from the mirror ATS that execute mirror test behaviour on the mirror components.

Table 2

Main Functionalities	Main Functionalities Mirror
AGPS	AGPS_mirror
GERAN	GERAN_mirror
HSU	HSU_mirror
Mac	Mac_mirror
RAB	RAB_mirror
RLC	RLC_mirror
RRC	RRC_mirror

5.2 PICS/PIXIT

To differentiate between the two approaches, the module parameter "pc_is_Intra" was introduced. If set to true, the Intra TTCN-3 Executable approach is used, if false, the Inter TTCN-3 Executable approach is used.

For each test cases, further 3GPP-specific sets of values for the further module parameters apply.

5.3 TTCN-3 coverage

The given ATS (reference ATS in combination with mirror ATS) covers 55 % of the TTCN-3 Core Language constructs in terms of an abstract syntax of the TTCN-3 Core Language standard. This coverage been determined by counting all constructs used in the ATS in relation to all available. Detailed measurements can be found in annex B.

5.4 The ATS in TTCN-3 core (text) format

The TTCN-3 core languages representation corresponding to this ATS is contained in ASCII files (the reference ATS in the folder ttcn3_reference, the mirror ATS in the folder ttcn3_mirror) in archive tr_102976v010201p0.zip which accompanies the present document.

Annex A: Definition of the Inter TTCN-3 Executable interface

For the Inter TTCN-3 Executable approach, a CoDec and a test adaptation need to be provided. In the present ATS, external functions have been replaced by TTCN-3 functions, thus a default Platform Adapter can be used. An example encoding and communication mechanism that have been used to validate the applicability of the Inter TTCN-3 are described in the following.

A.1 CoDec

The ATS is using three different kinds of messages that are sent over the TSI and therefore are of interest for the CoDec implementation. There are messages that are of types defined completely in TTCN-3, pure ASN.1 messages and messages of types defined in TTCN-3 that contain fields of types defined in ASN.1. For the pure ASN.1 messages the standardized PER Canonical Aligned encoding can be used.

For the TTCN-3 and mixed messages the proposed approach is to use the standardized XML encoding used for logging purposes as defined in ES 201 873-6 [i.4]. An example encoding that has been used to validate the applicability of the Inter TTCN-3 Executable approach follows:

TTCN-3 type definitions (value definitions not shown):

```
type union TheUnionType {
    TheRecordType theRecord
}
type record TheRecordType {
    integer theInteger,
    record of charstring theRecordOf
}
```

Corresponding XML representation:

```
<?xml version="1.0" encoding="UTF-8"?>
<Values:union module="TheModule" type="TheUnionType" name="">
   <Values:record module="TheModule" type="TheRecordType" name="theRecord">
       <Values:integer name="theInteger">
           <Values:value>5000</Values:value>
       </Values:integer>
       <Values:record_of module="TheModule" type="TheRecordOfType" name="theRecordOf">
           <Values:charstring name="0">
               <Values:value>192.168.89.90</Values:value>
           </Values:charstring>
           <Values:charstring name="1">
               <Values:value>192.168.89.91</Values:value>
           </Values:charstring>
       </Values:record_of>
   </Values:record>
</Values:union>
```

This usage of known standardized encodings for both the ASN.1 (PER) and TTCN-3 messages (generic TCI XML) is an approach that assures high probability of compatibility between different tools executing the ATS and the mirror ATS and is open for further extension of the test suite.

A.2 Test Adaptation

To validate the applicability of the Inter TTCN-3 Executable approach, the following simple communication mechanism has been used and realised in the System Adapters of the two involved test systems:

As communication channel, UDP is used. The involved IP addresses and port numbers need to be configurable in the System Adapters. Each TTCN-3 port defined in the component type “SystemInterface” is mapped to a single local UDP port.

As an alternative communication channel, TCP can used. The TCP System Adapter has to provide a client and a server mode to allow to be used as an adaptation of the reference test system as well as an adaptation of the mirror test system. Furthermore, The involved IP addresses and port number need to be configurable in the System Adapters. Each TTCN-3 port defined in the component type "SystemInterface" is mapped to a single TCP connection.

Since TCP is a stream based protocol additional knowledge is required in the System Adapter to assure that there are not more than one PDU in a TCP packet or a PDU is splitted over more than one TCP packet. As UDP does not have such a requirement, the preferred communication channel is UDP.

Annex B: Feature coverage details

Table B.1 shows the number of occurrences of TTCN-3 Core Language constructs together with a short description of the language construct. Constructs that are not used, are not listed.

Table B.1

TTCN-3 Construct	Number of occurrences	Description
create alive	35	create alive expression
all	28	all keyword
import all	386	import all
alt	419	alt statement
anonymous field	2 176	record of/array element
any	10	any keyword
Application	92 565	function call/indexing/field selection or quantified expression
ArrayDimension	18	array dimension in array specification
ArraySpec	18	array specification
Assignment	3 212	assignment
Attribute	19	single attribute
Attributes	19	attributes list
bitstring	4 480	bitstring literal
bitstring type	735	bitstring type reference
xor4b	6	xor operator
Block	4 472	statement block
boolean type	581	boolean type reference
case	4	case keyword
charstring	920	charstring literal
charstring literal	924	charstring literal or charstring pattern
charstring pattern	4	charstring pattern
charstring type	80	charstring type reference
charstring value	920	charstring value
component	10	component keyword for quantified expression
component definition	8	component type definition
Composed	64 929	& operation or masked expression
Configuration	57	map/connect/unmap/disconnect
Conjunction	60 376	and operator
connect	23	connect statement
const declaration	750	const declaration
Curly	8 827	field or list value
DeclarationKind	14	declaration kind for import of kind
declaration with	10 266	declaration with with block
default type	3	default type reference
Disjunction	58 632	or operator
Divide	44	/ operator
do while	9	do while statement
done	4	done event
else alternative	1	else alternative for alt statement
else case	2	else select case
encode	19	encode attribute
enum definition	368	enum type definition
Equal	3 790	operator
Expression	10 980	top level type or value expression
fail verdict	62	fail literal
false	691	false literal
Field	24 685	field of list or record or set or array
float type	8	float type reference
for	25	for statement
FormalPar	5 195	formal parameter
FormalPars	2 300	formal parameter list
FreeText	19	free text

TTCN-3 Construct	Number of occurrences	Description
FreeTextLine	97	line of free text
function	1 120	function definition
FunctionKind	10	function import declaration kind
Greater	46	> operator
GreaterOrEqual	25	> operator
group	38	group declaration
hexstring	92	hexstring literal
hexstring type	47	hexstring type reference
Identifier	136 342	identifier
if	911	if statement
ifpresent	240	ifpresent constraint
import	396	import statement
ImportBlock	14	import block
in	173	in parameter direction
inout	22	inout parameter direction
inconc verdict	47	Inconc verdict
inline alt	7	inline alt statement
inline template	64 889	inline template
InputAlternative	587	alternative for alt/interleave statement
InputEvent	586	input event for input alternative
InstanceDecl	3 031	declaration of one var/const/port/timer/modulepar
integer usage	453	integer literal
integer type definition	2 561	integer type reference
Invocation	5 104	function call statement
killed	4	killed event
KindImport	10	import with declaration kind
LValue	3 316	left hand side of assignment or target of >
language	78	language clause for module and import statement
length specification	449	Length specification
Less	42	< operator
LessOrEqual	13	< operator
mtc	12	mtc reference
Map	34	map statement
Masked	64 990	4b operator or inverted expression
message	32	message port type kind
Minus	147	subtraction operator or negation operator
Modal	64 891	ifpresent expression or modified expression
modifies	121	modified expression
module	45	module definition
ModuleBodyDeclaration	9 932	top level module declaration
Modules	396	module references in import statement
Modulo	79	mod operator
Multiply	64	* operator
NamedValue	2 721	enum value declaration
Negation	78	not operator
NotEqual	66	! operator
Null	68	null keyword
objid type	3	objid type reference
octetString	672	octet string literal
octetString type definition	169	octetstring type reference
omit	5 048	omit keyword
optional	6 009	optional field declaration
out	172	out parameter direction
Output	437	output event (send, raise, reply, call)
ParamsDeclaration	2	modulepar declaration
Paren	16 767	( ) operator
pass verdict	301	pass literal
Plus	205	+ operator
port declaration	80	port declaration
port definition	32	port type definition
PortInterfaceDef	326	port type interface (body)
PortRef	114	port reference for configuration statement
Product	65 280	product operation or signed expression

TTCN-3 Construct	Number of occurrences	Description
Quantified	38	any or all expression (any port, any component, any timer)
Question	724	? wildcard
range	800	.. constraint
receive	510	receive event
record	3 409	record type definition
Relation	60 947	relation operation or composed expression
Restricted	64 891	length restricted operation or primitive expression
return	400	return statement
returns	250	return clause
runs on	1 099	runs on clause
Scalar	6 701	number
select	2	select statement
select case	2	select statement case
self	35	self reference
send	437	send statement
set	1	set type definition
Shifted	64 996	shifted operation or arithmetic expression
Signed	68	signed operation or application expression
Square	399	[ ] operator
star	487	* wildcard
StructDef	3 613	record, union or set type definition
StructFieldDef	12 210	field declaration for record, union or set type
StructOfDef	427	record of or set of type definition
SubtypeDef	1 111	subtype definition
SubtypeSpec	5 084	subtype specification
Sum	64 996	sum operation or product expression
system	34	system reference
SystemSpec	20	system specification/system clause
template	1 050	template modifier
TemplateDef	2 040	template definition
TemplateKind	2	template modifier kind
testcase	20	testcase definition
Teststep	9	altstep definition
timeout	68	timeout event
timer	44	timer keyword
timer declaration	89	timer declaration
true	558	true literal
type	25 576	type reference
TypeKind	2	type import declaration kind
TypedDecl	2 914	multi declaration of same type (var, const, port, modulepar)
union	630	union declaration
Value	104	value assignment clause for getreply event
VarDecl	1 875	var declaration
while	10	while statement
with	19	with block

Annex C: Using checkpoints to trace the executed paths in the reference ATS

As described in clause 4.1.4, a mirror test case may intentionally drive a reference test case into a path where a non **pass** verdict is set. Thus, this verdict does not indicate that the TTCN-3 tool assurance test has failed or is inconclusive - instead a detailed analysis of each test run is required. While the test cases of the ATS described in clause 5 require to analyse which path of the reference ATS was actually executed, the following clauses outline a more advanced approach based on check points:

Check points can be introduced at each branch of the reference ATS (e.g. by `log` statements) to allow the analysis which branches of the reference ATS were executed. Each execution path relates to a particular sequence of check points passed through. By using a unified log format for check points, even a tool-based automatic evaluation of the correct sequence of check points passed through is possible.

Through the introduction of statements for generating check points, the reference ATS needs to be changed and may as a consequence not be reasonable usable as a conformance test suite for a real SUT anymore. However, this cannot be avoided because it is a different application if an ATS drives a real SUT for conformance testing of the SUT and if an ATS is driven by a mirror ATS for conformance testing of TTCN-3 tools. In addition, further changes of the reference ATS may anyway be necessary depending on the chosen back-to-back test approach as described in the next clauses.

Depending on whether the Intra TTCN-3 Executable approach or the Inter TTCN-3 Executable approach is used, it may be considered to even remove all verdict setting TTCN-3 statements from the reference ATS and to rely on the check points only:

- In the Inter TTCN-3 Executable approach, each test system maintains its own global test verdict. Thus, the global verdict of the test system executing the reference ATS may be simply ignored, i.e. verdict setting TTCN-3 statements might be kept in the reference ATS, but not considered. Instead, the correct order of passed through check points is considered.
- In the Intra TTCN-3 Executable approach, the test verdict of the reference ATS and the mirror ATS get mixed, thus it is not possible to consider only the verdict of a mirror test case. Hence, it may be considered to remove all verdict setting statements from the reference ATS and to consider only the control flow as indicated by the introduced check points.

History

Document history		
V1.1.1	June 2009	Publication
V1.2.1	December 2009	Publication