ETSI TS 132 322 V12.1.0 (2020-08)

TECHNICAL SPECIFICATION

Digital cellular telecommunications system (Phase 2+) (GSM);
Universal Mobile Telecommunications System (UMTS);
LTE;
Telecommunication management;
Test management Integration Reference Point (IRP);
Information Service (IS)
(3GPP TS 32.322 version 12.1.0 Release 12)

Reference

RTS/TSGS-0532322vc10

Keywords

GSM,LTE,UMTS

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

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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 (https://ipr.etsi.org/).

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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x   the first digit:

1   presented to TSG for information;

2   presented to TSG for approval;

3   or greater indicates TSG approved document under change control.

y   the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z   the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Test management Integration Reference Point (IRP), as identified below:

32.321:   "Test management Integration Reference Point (IRP); Requirements"

32.322:   "Test management Integration Reference Point (IRP): Information Service (IS)"

32.326:   "Test management Integration Reference Point (IRP): Solution Set (SS) definitions"

A 3G telecommunication network is composed of a multitude of different Network Elements (NE). For a successful operation of the network the operator must be provided with mechanisms allowing him to manage the network. These management activities can be grouped into several areas: configuration management, fault management, performance management, accounting management and security management.

A management function assisting in different high level management areas such as fault management and performance management is test management. The purpose of testing is to get information about the functionality and performance of the 3G managed network subject to the test.

The present document is part of a TS-family defining the Telecommunication Management (TM) of 3G systems. The TM principles are described in 3GPP TS 32.101 [5]. The TM architecture is described in 3GPP TS 32.102 [6]. The other specifications define the interface (Itf-N) between the managing system (manager), which is in general the Network Manager (NM) and the managed system (agent), which is either an Element Manager (EM) or the managed NE itself. The Itf-N is composed of a number of integration reference points (IRPs) defining the information in the agent that is visible for the manager, the operations that the manager may perform on this information and the notifications that are sent from the agent to the manager. One of these IRPs is the Test Management IRP.

Each IRP is specified by the requirements part, the IS part and at least one SS (e.g. CORBA SS).

1 Scope

The present document defines the IS part of the Test Management IRP, which describes the semantics of the information and the interactions visible across Itf-N in a protocol independent way. The information is specified by means of information object classes and the interactions by means of operations and notifications. The present document does not specify the syntax (encoding) of the information.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

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

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1]   3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".

[2]   3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[3]   3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[4]   ITU-T Recommendation X.733: "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[5]   ITU-T Recommendation X.745: "Information technology - Open Systems Interconnection - Systems Management: Test management function".

[6]   3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[7]   3GPP TS 32.102: "Telecommunication management; Architecture".

[8]   3GPP TS 32.321: "Telecommunication management; Test management Integration Reference Point (IRP): Requirements".

[9]   3GPP TS 32.672: "Telecommunication management; Configuration Management (CM); State Management Integration Reference Point (IRP): Information Service (IS)".

[10]   ITU-T Recommendation X.737: "Information technology - Open Systems Interconnection - Systems Management: Confidence and diagnostic test categories.

[11]   3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[12]   IETF RFC4656: "A One-way Active Measurement Protocol (OWAMP)".

[13]   IETF RFC 5357: "A Two-way Active Measurement Protocol (TWAMP)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [6], 3GPP TS 32.102 [7] and 3GPP TS 32.321 [8] apply.

3.2 Abbreviations

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

EM	Element Manager
IOC	Information Object Class
IRP	Integration Reference Point
IS	Information Service
ME	Element Manager
MORT	Managed Object Referring to Test
NE	Network Element
TM	Telecommunication Management
TO	Tester Object
VSE	Vendor-Specific-Extended
VSE TO	Vendor-Specific-Extended Tester Object

4 System Overview

4.1 System Context

The general definition of the System Context for the present IRP is found in 3GPP TS 32.150 [11] subclause 4.7.

In addition, the set of related IRP(s) relevant to the present IRP is shown in the two diagrams below.
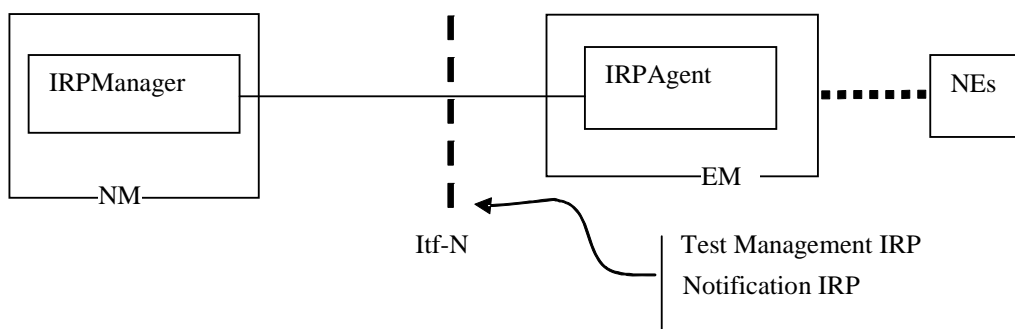
Figure 1: System Context A
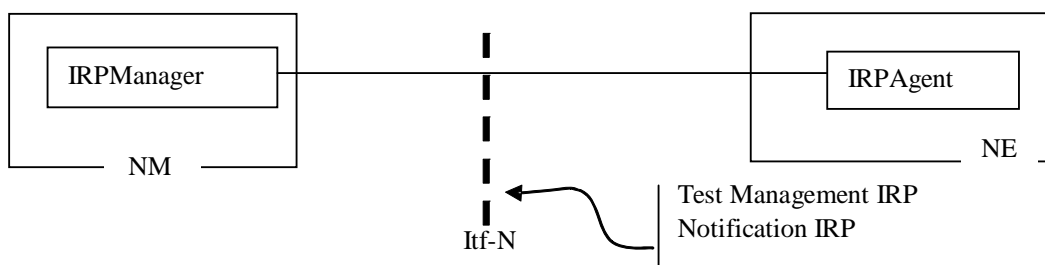
Figure 2: System Context B

5 Information Object Classes (IOCs)

5.1 Information Entities imported and local Labels

Label reference	Local label
3GPP TS 32.622 [3], information object class, Top	Top
3GPP TS 32.622 [3], information object class, IRPAgent	IRPAgent
3GPP TS 32.312 [2], information object class, managedGenericIRP	managedGenericIRP

5.2 Class Diagram

5.2.1 Attributes and Relationships

The following figure shows, for the Test Management IRP, the class definitions and the relationships between the classes.
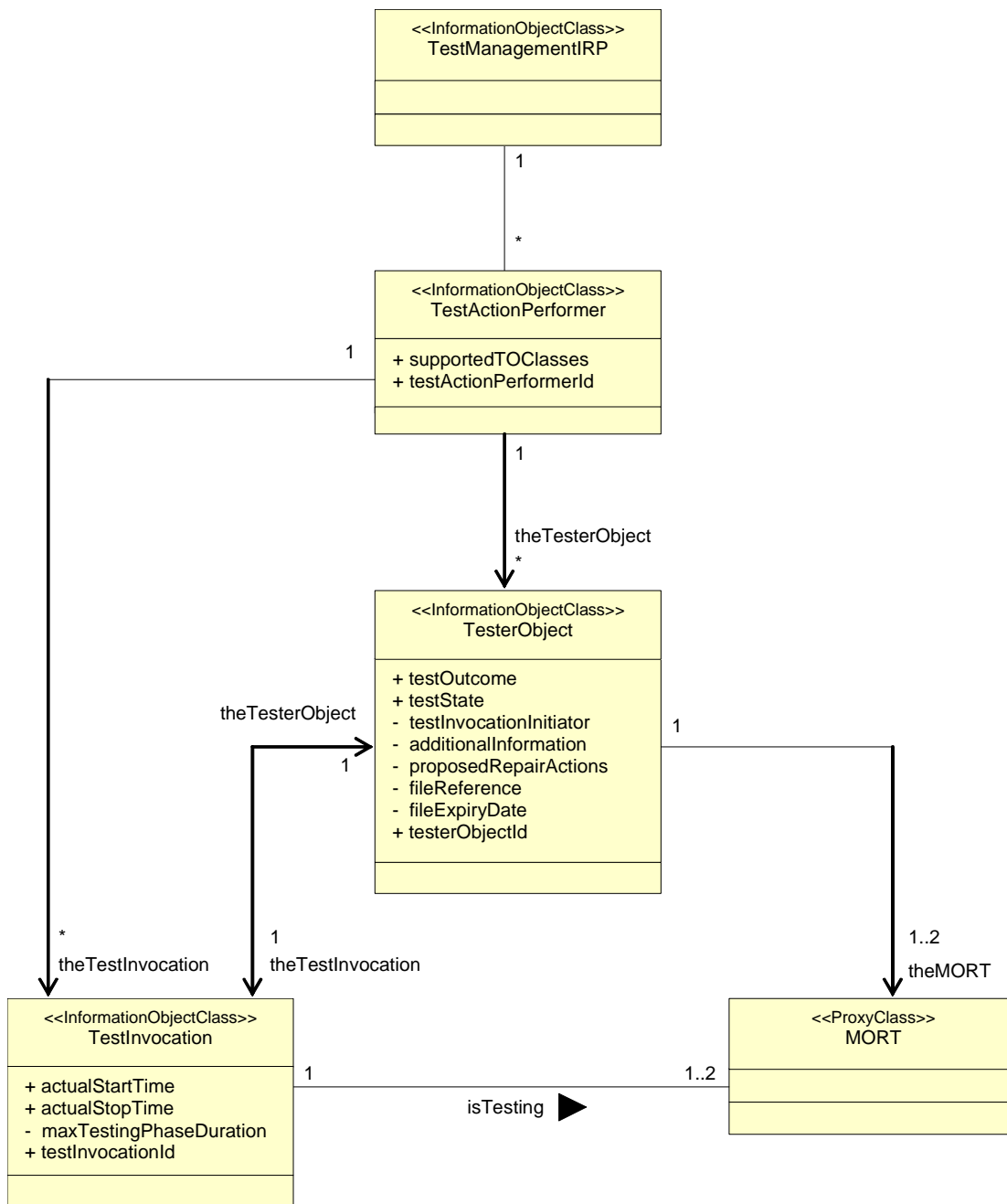
From the cardinalities can be seen that each instance of *TestManagementIRP* may have several instances of *TestActionPerformer*. Each instance of *TestActionPerformer* can have multiple instances of associated *TesterObjects*. Each instance of *TesterObject* in turn has one instance of *TestInvocation*. For the resource self test, each instance of TesterObject has one instance of MORT. For the connection and loopback test, each instance of TesterObject has two instances of MORT.

5.2.2 Inheritance

The following figure depicts the inheritance relationships between the information object classes.
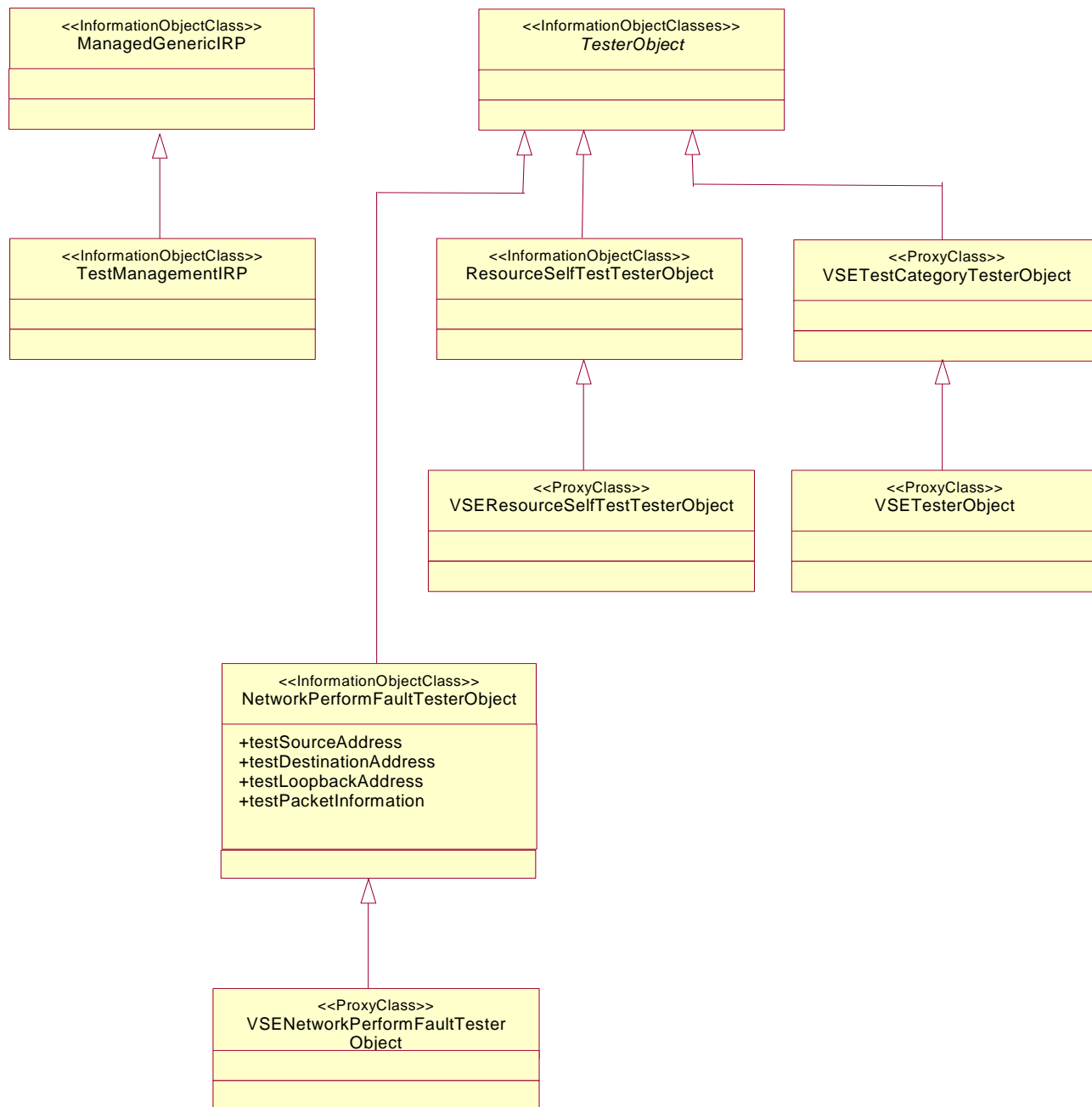
5.3 Information Object Classes (IOCs) definition

5.3.1 Information Object Class *TestManagementIRP*

5.3.1.1 Definition

The IOC *TestManagementIRP* together with the IOC *TestActionPerformer* represent the test management capabilities defined by this specification. To conduct a test of network resources, this object may require capabilities of other

objects such as *TesterObject*. The IOC *TestManagementIRP* inherits from the IOC *ManagedGenericIRP* specified in 3GPP TS 32.312 [2].

5.3.1.2 Attributes

The IOC *TestManagementIRP* has no own attributes, only those inherited from the IOC *ManagedGenericIRP*.

5.3.2 Information Object Class *TestActionPerformer*

5.3.2.1 Definition

The IOC *TestActionPerformer* provides the ability to receive and react upon test requests. This class must also be able to instantiate and delete tester objects or, in case the tester objects are permanently instantiated, to allocate and reserve them for their usage. This specification does not require this IOC to be instantiated. It may be abstract and used for inheritance purposes only. In this way the ability to receive and react upon test requests may be included in any other IOC.

5.3.2.2 Attributes

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
supportedTOClasses	+	M	M	-
testActionPerformerId	+	M see note	M	-
NOTE: This attribute is only mandatory in case the IOC TestActionPerformer is instantiated. In case this IOC is an abstract class and used for inheritance purposes only the attribute shall be omitted.				

5.3.3 Information Object Class *TesterObject*

5.3.3.1 Definition

The IOC *TesterObject* monitors and controls the testing of a *MORT* instance and reports the outcome of the test execution. Tester Objects (TOs) are instantiated by the IOC *TestActionPerformer* in response to a valid test initiation request (*initiateTests*). They are deleted after termination of the test. It is also possible that TOs are permanently instantiated. In this case they are allocated to a certain *TestActionPerformer* during the test execution. After termination of the test they are released.

The IOC *TesterObject* defines a generic TO. It shall be used as an abstract class from which more specific tester objects shall be derived by specialisation for each test category. Test categories and the associated test category specific TOs are defined in ITU-T Recommendation X.737 [10]. These test category specific TOs can be specialised further by defining Vendor-Specific-Extended (VSE) TOs. The generic TO defines attributes pertaining to a test and required for all test categories.

Each test invocation shall have only one associated TO.

Only test category specific TOs or VSE TOs shall be instantiated.

For simplicity this specification will often use only the term TO. In this case either the test category specific TO or the VSE TO is referred to depending on which is actually instantiated.

5.3.3.2 Attributes

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
testOutcome	+	M	M	-
testState	+	M	M	-
testInvocationInitiator	-	C	M	-
additionalInformation	-	O	-	-
proposedRepairActions	-	O	-	-
fileReference	-	M see note	-	-
fileExpiryDate	-	M see note	-	-
testerObjectId	+	M	M	-
NOTE: In case the TO does support capturing test results in a file this parameter shall be present and contain information. In case the TO does not support capturing test results in a file this parameter shall contain no information or shall be absent.				

5.3.4 Information Object Class *ResourceSelfTestTesterObject*

5.3.4.1 Definition

The IOC *ResourceSelfTesterObject* is a specialised TO for the resource self test. It inherits from the IOC *TesterObject*. It specifies the triggering events for the emission of the test result notifications.

5.3.4.2 Attributes

This IOC has no own attributes, only those inherited from the generic IOC *TesterObject*.

5.3.5 Proxy Class VSETestCategoryTesterObject

5.3.5.1 Definition

Certain tests may not fit in any of the test categories defined in ITU-T Recommendation X.737 [10]. In this case vendors may define new (VSE) test categories and the associated test category specific TOs. The Proxy Class *VSETestCategoryTesterObject* represents the set of these VSE test category tester objects

The IOCs represented by the Proxy Class *VSETestCategoryTesterObject* shall inherit from the IOC *TesterObject*.

> NOTE: A vendor may also claim 3GPP compliance to a certain release in case that a specific test fits into one of the ITU-T test categories without that the corresponding ITU-T test category specific TO is supported in this release supposed that this test category specific TO will be added in a later release than the current one. The vendor shall update this specification in due time.

5.3.5.2 Attributes

The attributes of this IOC are vendor specific.

5.3.6 Proxy Class VSEResourceSelfTestTesterObject

5.3.6.1 Definition

In case the IOC *ResourceSelfTestTesterObject* does not fulfil the specific requirements of a certain resource self test, vendors may define proprietary IOCs by further specialisation. The Proxy Class *VSEResourceSelfTestTesterObject* represents these IOCs.

The IOCs represented by the Proxy Class *VSEResourceSelfTestTesterObject* shall inherit from the IOC *ResourceSelfTestTesterObject*.

5.3.6.2 Attributes

Apart from the attributes inherited the attributes of the IOCs represented by this Proxy Class are vendor specific.

5.3.7 Proxy Class *VSETesterObject*

5.3.7.1 Definition

In case an IOC represented by the Proxy Class *VSETestCategoryTesterObject* does not fulfil the specific requirements of a certain test, vendors may define proprietary IOCs by further specialisation. The Proxy Class *VSETesterObject* represents these IOCs.

The IOCs represented by the Proxy Class *VSETesterObject* shall inherit from the associated IOC represented by the Proxy Class *VSETestCategoryTesterObject*.

5.3.7.2 Attributes

Apart from the attributes inherited the attributes of the IOCs represented by this Proxy Class are vendor specific.

5.3.8 Proxy Class *MORT*

5.3.8.1 Definition

The ProxyClass *MORT* represents a network resource that is under test. Its class definition shall be one defined in the various 3GPP Network Resource Model specifications or defined by a VSE network resource class.

5.3.8.2 Attributes

This IOC has no attributes.

5.3.9 Information Object Class *TestInvocation*

5.3.9.1 Definition

The IOC *TestInvocation* is the abstract representation of a test invocation. A test invocation shall aim to test one or more capabilities of a MORT. The IRPManager can request for the establishment of a test invocation using the operation *initiateTests*.

A MORT can be complex in that there are multiple capabilities that can be subject to test. Therefore, it is possible to have multiple test activities active, all aimed at the same MORT but on its different capabilities. Whether multiple test activities can be testing the same MORT capabilities at the same time is an implementation decision, probably based on the nature and behaviour of the TO, and therefore, is outside the scope of this specification.

5.3.9.2 Attributes

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
actualStartTime	+	O	M	-
actualStopTime	+	O	M	-
maxTestingPhaseDuration	-	O	-	-
testInvocationId	+	M	M	-

5.3.10 Information Object Class *NetworkPerformFaultTesterObject*

5.3.10.1 Definition

The IOC NetworkPerformFaultTesterObject is a specialised TO for the network performance test. It inherits from the IOC TesterObject.

5.3.10.2 Attributes

Apart from the attributes inherited from the generic IOC `TesterObject`, this IOC has own attributes in the below table.

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
testSourceAddress	+	M	M	M
testDestinationAddress	+	M	M	M
testLoopbackAddress	+	M	M	M
testPacketInformation	+	M	M	M

5.3.11 Proxy Class VSENetworkPerformFaultTesterObject

5.3.11.1 Definition

In case the IOC `NetworkPerformFaultTesterObject` does not fulfil the specific requirements of a certain connection test, vendors may define proprietary IOCs by further specialisation. The Proxy Class `VSENetworkPerformFaultTesterObject` represents these IOCs.

The IOCs represented by the Proxy Class `VSENetworkPerformFaultTesterObject` shall inherit from the IOC `NetworkPerformFaultTesterObject`.

5.3.11.2 Attributes

Apart from the attributes inherited the attributes of the IOCs represented by this Proxy Class are vendor specific.

5.4 Information relationships definition

5.4.1 Relationship between TestManagementIRP and TestActionPerformer

5.4.1.1 Definition

This relationship defines a binary association between the IOC *TestManagementIRP* and the IOC *TestActionPerformer*.

5.4.1.2 Roles

This relationship has no roles.

5.4.2 Relationship between TestActionPerformer and TesterObject

5.4.2.1 Definition

This relationship defines a binary association between the IOC *TestActionPerformer* and the IOC *TesterObject*. The association is navigable from the *TestActionPerformer* to the *TesterObject*.

5.4.2.2 Roles

Name	Definition
theTesterObject	This rolename provides a name allowing to navigate from an instance of *TestActionPerformer* to the associated instances of *TesterObject*. If *tap* is an instance of *TestActionPerformert*, the expression *tap.theTesterObject* yields the set of object instances of *TesterObject*.

5.4.3 Relationship between TestActionPerformer and TestInvocation

5.4.3.1 Definition

This relationship defines a binary association between the IOC *TestActionPerformer* and the IOC *TestInvocation*. The association is navigable from the *TesterObject* to the *TestInvocation*.

5.4.3.2 Roles

Name	Definition
theTestInvocation	This rolename provides a name allowing to navigate from an instance of *TestActionPerformer* to the associated instances of *TestInvocation*. If *tap* is an instance of *TestActionPerformert*, the expression *tap.theTestInvocation* yields the set of object instances of *TestInvocationt*.

5.4.4 Relationship between *TesterObject* and *TestInvocation*

5.4.4.1 Definition

This relationship defines a binary association between the IOC *TesterObject* and the IOC *TestInvocation*. The association is navigable in both directions.

5.4.4.2 Roles

Name	Definition
theTesterObject	This rolename provides a name allowing to navigate from an instance of *TestInvocation* to the associated instance of *TesterObject*. If *ti* is an instance of *TestInvocation*, the expression *ti.theTesterObject* yields an object instance of *TesterObject*.
theTestInvocation	This rolename provides a name allowing to navigate from an instance of *TesterObject* to the associated instance of *TestInvocation*. If *to* is an instance of *TesterObject*, the expression *to.theTestInvocation* yields an object instance of *TestInvocation*.

5.4.5 Relationship between *TesterObject* and *MORT*

5.4.5.1 Definition

This relationship defines a binary association between the IOC *TesterObject* and the Proxy Class *MORT*. The association is navigable from the *TesterObject* to the *MORT*.

5.4.5.2 Roles

Name	Definition
theMORT	This rolename provides a name allowing to navigate from an instance of *TesterObject* to the associated instance of *MORT*. If *to* is an instance of *TesterObject*, the expression *to.theMORT* yields an object instance of *MORT*.

5.4.6 Relationship between *TestInvocation* and *MORT*

5.4.6.1 Definition

This relationship defines an association between the IOC *TestInvocation* and the IOC *MORT*. This association specifies that the latter is testing the former.

5.4.6.2 Roles

Instead of roles this relationship has a role name.

5.5 Information attributes definition

5.5.1 Definition and legal Values

Attribute Name	Definition	Legal Values
testInvocationId	This attribute uniquely identifies an instance of a *TestInvocation* within the TestManagementIRP. The test invocation identifier is assigned by the TestActionPerformer. When a testInvocationId can be reused is outside the scope of this specification.	
testState	This attribute reflects the actual test state (ITU-T Recommendation X.745 [5]).	ENUM {notInitialized, idle, initializing, testing, terminating, disabled}
testOutcome	This attribute provides information about the test result, as perceived by the associated TO, in a standardised manner. The information in this parameter is only valid after termination of the test activity. This information shall be present in the last test result notification emitted by a TO prior to its deletion.	ENUM {pass, fail, inconclusive, timed-out, premature-termination} Pass indicates that the test exercise of the test invocation has executed correctly and has found no problem. Fail indicates that the test exercise of the test invocation has executed correctly and has found one or more problems. Inconclusive indicates that the TO has not determined if the execution is Pass or Fail. Timed-out indicates that the TO has terminated its execution because of the expiry of the timer (i.e., the current time – TestSession.sessionStartTime >= TesterObject.timeOut). Premature termination indicates that the TO has (a) never started execution or (b) terminated its execution prematurely, either by *TestManagementIRP* and its associated objects internal problems or in response to a *terminateTests* operation.
supportedTOClasses	This attribute identifies the TO classes that are supported by a certain managed object instance whose class has inherited from *TestActionPerformer* or whose class is the *TestActionPerformer*.	SET OF TO class name
testActionPerformerId	This attribute unambiguously identifies an instance of a *TestActionPerformer*.	
testerObjectId	This attribute unambiguously identifies an instance of a *TesterObject*.	
testInvocationInitiator	It identifies the IRPManager.	How multiple IRPManagers choose their identifier so that they are distinguishable is outside the scope of this specification.
additionalInformation	This attribute holds a set of additional information pertaining to the test.	The semantics of this parameter are outside the scope of this specification
proposedRepairActions	This attribute suggests one or more repair actions if the reason for a failure is known.	The semantics of this parameter are outside the scope of this specification
actualStartTime	This attribute specifies the time at which the TO will enter or has entered the test state *testing*. Before the TO enters the testing state this is an estimated time. After entering the testing state this is the actual time. Note that this is not the time of the invocation of the operation *initiateTests*.	All values indicating a valid time.
actualStopTime	This attribute specifies the time at which the TO will leave or has left the test state *testing*. Before the TO leaves the testing state this is an estimated time. After leaving the testing state this is the actual time. Note that this is not the time of the invocation of the operation *terminateTests*.	All values indicating a valid time later than the *actualStartTime*.

Attribute Name	Definition	Legal Values
maxTestingPhaseDuration	This attribute specifies the maximum amount of time that a TO may spend in the testing state.	All values indicating a valid amount of time.
fileReference	This attribute carries the reference to a file that contains the test result data set.	
fileExpiryDate	This attribute carries the date and time after which the file, whose reference is carried by the *fileReference* attribute, may be removed.	All values indicating a valid time.
testSourceAddress	This attribute provides the source address of the test.	An IP address indicating source.
testDestinationAddress	This attribute provides the destination address of the test.	An IP address indicating destination. In a loopback test, it should be NULL.
testLoopbackAddress	This attribute provides the loopback address in a loopback test.	An IP address indicating loopback point in a loopback test. In connection test, it should be NULL.

6 Interface definition

6.1 Class diagram representing interfaces

The following diagram depicts the interfaces of the Test Management IRP with their corresponding operations and notifications.
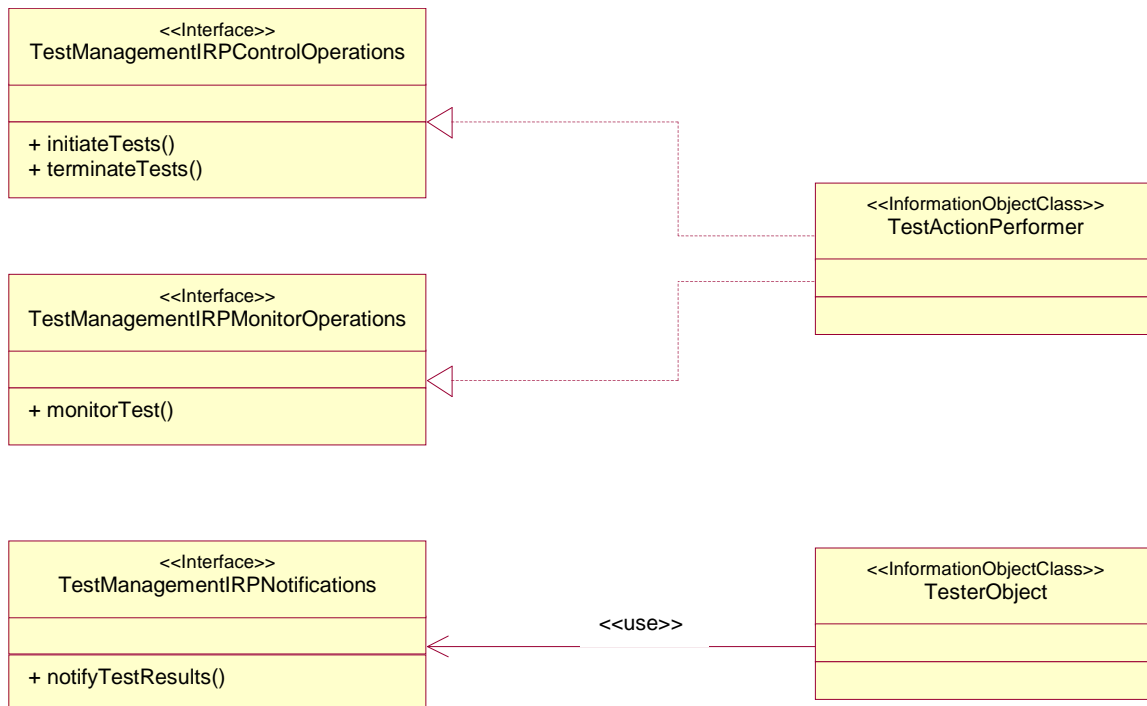
6.2 Generic rules

Rule 1: Each operation with at least one input parameter supports a pre-condition valid_input_parameter which indicates that all input parameters shall be valid with regards to their information type. Additionally, each such operation supports an exception operation_failed_invalid_input_parameter which is raised when pre-condition valid_input_parameter is false. The exception has the same entry and exit state.

Rule 2: Each operation with at least one optional input parameter supports a set of pre-conditions supported_optional_input_parameter_xxx where "xxx" is the name of the optional input parameter and the pre-condition indicates that the operation supports the named optional input parameter. Additionally, each such operation supports an exception operation_failed_unsupported_optional_input_parameter_xxx which is raised when (a) the pre-condition supported_optional_input_parameter_xxx is false and (b) the named optional input parameter is carrying information. The exception has the same entry and exit state.

Rule 3: Each operation shall support a generic exception operation_failed_internal_problem that is raised when an internal problem occurs and that the operation cannot be completed. The exception has the same entry and exit state.

6.3 Interface testManagementIRPControlOperations

The interface *TestManagementIRPControlOperations* contains the operations *initiateTests* and *terminateTests*. It must be implemented by every object with the ability to receive and react upon test requests, for example by every instance of *TestActionPerformer*.

6.3.1 Operation *initiateTests* (M)

6.3.1.1 Definition

The IRPManager uses this operation to request the IRPAgent to initiate controlled tests. A single test request may initiate multiple (one or more) tests.

For each test to be initiated the managed object representing the network resource to be tested and the tester object class must be specified.

The initiated tests are independent and not related to each other. This implies that independent test result notifications are sent for each of the tests initiated by s single *initiateTests* operation.

6.3.1.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
testInvocationInitiator	C	TesterObject. testInvocationInitiator	This parameter identifies the IRPManager..

toBeInitiatedTests	M	SET OF SET {
		    maxTestingStateDuration (O)
		    toBeTestedMORT (O)
		    testerObjectClass (M)
		    testerObjectName (O)
		    testerObjectInitialAttributeList (O)
		}
			This sequence specifies the tests to be initiated.
			For each test the parameter *maxTestingStateDuration* specifies the timeout period of the tests to be initiated. One value shall indicate "No limit".
			toBeTestedMORT specifies the instance of the *MORT* to be tested. If the parameter is absent, the *MORT* is identical to the object instance to which the subject operation is directed to.
			The parameter *testerObjectClass* specifies the class of the associated tester object. Optionally, a name for the tester object instance may be specified in the parameter *testerObjectName*.
			The parameter *testerObjectInitialAttributeList* carries some or all the values of the attributes of the TO instance responsible for the test. The syntax and semantics of this attribute value is dependent on the specific TO class definition and is outside the scope of 3GPP.

6.3.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment

response	M	SEQUENCE OF CHOICE {
		    testInitiated
		    testNotInitiated
		}
		
		testInitiated = SEQUENCE {
		    TestInvocation.testInvocationId (M)
		    testerObjectName (O)
		}
		
		testNotInitiated =
		failureReason
			The number and the order, related to the tests to be initiated, of elements in this sequence and in the set of the input parameter *toBeInitiatedTests* shall be identical.
			For a successfully instantiated test the parameter *testInitiated* returns the test invocation identifier of the test. In case the tester object name has not been specified in the request it shall be returned in *testerObjectName*.
			For a failed test instantiation the parameter *testNotInvoked* returns the reason why the instantiation of the test failed.
			Failure reasons are
			TO class is not existing
			MORT is not existing
			MORT is not available
			others

6.3.1.4 Pre-condition

The precondition must hold true before the operation is invoked. The pre-condition depends on the test category.

For at least one of the specified tests to be instantiated the following must hold true:

theIndicatedMORTIsExisting **AND** theIndicatedMORTIsAvailable **AND** theIndicatedTOClassIsExisting.

Assertion Name	Definition
theIndicatedMORTIsExisting	The MORT indicated by the subject operation for this test is existing
theIndicatedMORTIsAvailable	The MORT indicated by the subject operation for this test is available.
theIndicatedTOClassIsExisting	The TO class indicated by the subject operation for this test is existing.

6.3.1.5 Post-condition

The post-condition must hold true after the completion of the operation:

allIndicatedTOsInstantiated **OR** notAllTestsInitiated **OR** noTestInitiated

Assertion Name	Definition
allTestsInitiated	All tests indicated by the subject operation were initiated successfully.
notAllTestsInitiated	Not all but at least one test indicated by the subject operation was initiated successfully.
noTestInitiated	No test indicated by the subject operation was initiated successfully.

6.3.1.6 Exceptions

Exception Name	Definition
operationFailedEntirely	**Condition:** noTestInitiated = TRUE **Returned information:** The response parameter is returned **Exit state:** Entry state
operationFailedPartly	**Condition:** notAllTestsInitiated = TRUE **Returned information:** The response parameter is returned **Exit state:** Entry state

6.3.2 Operation *terminateTests* (M)

6.3.2.1 Definition

The IRPManager uses this operation to request the IRPAgent to terminate tests during their life time. A single *terminateTests* operation may terminate multiple (one or more) tests.

The tests to be terminated are identified by their test invocation identifiers. The IRPManager terminating a test may be different from the IRPManager that initiated the test. The *terminateTests* operation must be invoked on the object which received the corresponding *initiateTests* operation.

6.3.2.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
toBeTerminatedTests	M	SET OF TestInvocation.testInvocationId	This parameter specifies the tests that shall be terminated.

6.3.2.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment

response	M	SEQUENCE OF CHOICE {
		    testTerminated
		    testNotTerminated
		}
		testTerminated =
		TestInvocation.testInvocationId
		testNotTerminated =
		SEQUENCE {
		    TestInvocation.testInvocationId,
		    failureReason
		}
			The number and the order, related to the test invocation identifier, of elements in this sequence and in the set of the input parameter *toBeTerminatedTests* shall be identical.
			It specifies the test invocation ids of the tests, that were successfully terminated, and the ids of the tests, that failed to be terminated successfully together with the reason for the failure.
			Failure reasons are
			test invocation id is not existing
			others

6.3.2.4 Pre-condition

The precondition must hold true before the operation is invoked.

allIndicatedTestInvocationIdsAreExisting **OR** notAllIndicatedTestInvocationIdsAreExisting

Assertion Name	Definition
allIndicatedTestInvocationIdsAreExisting	All test invocation identifiers specified by the subject operation are existing.
notAllIndicatedTestInvocationIdsAreExisting	Not all but at least one test invocation identifier specified by the subject operation is existing.

6.3.2.5 Post-condition

The post-condition must hold true after the completion of the operation.

allIndicatedTestsTerminated **OR** notAllIndicatedTestsTerminated **OR** noIndicatedTestTerminated

Assertion Name	Definition
allIndicatedTestsTerminated	All tests indicated in the subject operation were terminated successfully.
notAllIndicatedTestsTerminated	Not all but at least one test indicated in the subject operation aaws terminated successfully
noIndicatedTestTerminated	No test indicated in the subject operation aaws terminated successfully

6.3.2.6 Exceptions

Exception Name	Definition
operationFailedEntirely	**Condition:** noIndicatedTestTerminated = TRUE **Returned information:** The response parameter is returned **Exit state:** Entry state
operationFailedPartly	**Condition:** notAllIndicatedTestInvocationIdsAreExisting = TRUE **OR** notAllIndicatedTestsTerminated = TRUE **Returned information:** The response parameter is returned **Exit state:** Entry state

6.4 Interface TestManagementIRPMonitorOperations

The interface *TestManagementIRPMonitorOperations* contains the operation *monitorTest*. It has a realisation relationship with the IOC *TestActionPerformer*.

6.4.1 Operation *monitorTest* (M)

6.4.1.1 Definition

IRPManager shall be able to retrieve information about the test as observed by the TO during the test execution by reading the relevant attributes of the TO associated to the test. Also after the test execution the manager shall be able to read these attributes as long as the TO exists. Attributes conveying information about the test execution are *testState* and *testOutcome*. Depending on the specific test category specific TO or the VSE TO other attributes may also contain information about the test execution. In this case the subject operation may also allow to read the values of these attributes.

6.4.1.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
toBeMonitoredTO	M	tOInstance	This parameter specifies the instance of the tester object, whose attribute values of *testState*, *testOutcome* and other attributes shall be retrieved.

6.4.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
monitoredAttributeValues	M	SET { TesterObject.testState (M) TesterObject.testOutcome (M) other attributes (O) }	This parameter shall be returned if all attributes were read successfully and may be returned, if at least one attribute was read successfully. The values to be returned are those prevalent at the time of the reception of the subject operation.
error	M	failureReason	This parameter shall be returned if the specified tester object instance is not existing or, in case the tester object instance is existing, at least one attribute could not be read, i. e. if operationFailedEntirely = TRUE OR operationFailedPartly = TRUE The parameter returns the failure reason.

6.4.1.4 Pre-condition

The precondition must hold true before the operation is invoked.

indicatedTOInstanceIsExisting

Assertion Name	Definition
toBeMonitoredTOIsExisting	The TO instance indicated by the subject operation is existing.

6.4.1.5   Post-condition

The post-condition must hold true after the completion of the operation.

allAttributeValuesRead **OR** notAllAttributeValuesRead **OR** noAttributeValueRead

Assertion Name	Definition
allAttributeValuesRead	All attributes of the TO indicated by the subject operation were read successfully.
notAllAttributeValuesRead	Not all but at least one attribute of the TO indicated by the subject operation were read successfully.
noAttributeValueRead	No attribute of the TO indicated by the subject operation was read successfully.

6.4.1.6   Exception

Exception Name	Definition
operationFailedEntirely	**Condition:** toBeMonitoredTOIsExisting = FALSE **OR** noAttributeValueRead = TRUE **Returned information:** The error parameter returns the object identifier of the TO that does not exist or the reasons, why the attributes could not be read **Exit state:** Entry state
operationFailedPartly	**Condition:** toBeMonitoredTOIsExisting = TRUE **AND** notAllAttributeValuesRead = TRUE **Returned information:** The error parameter returns the reason why an attribute could not be read. The attribute that could be read my be returned in the parameter *error* or the parameter *attributeList* **Exit state:** Entry state

6.5 Interface TestManagementIRPNotifications

6.5.1 Notification *notifyTestResults* (M)

6.5.1.1 Definition

Test results are made available to the IRPManager by one or more notifications *notifyTestResults* emitted by the TO that is related to the test invocation.

Depending on the nature of the test and the specification of the TO behaviour, the TO may need to convey to the IRPManager a test result data set. There are two ways to convey this kind of information. One way is to use the parameter *additionalInformation* of the notification. In this case, the *fileReference* and *fileExpiryDate* shall contain no information or be absent. The other way is to use a file to capture the test result data set. In this case, the *additionalInformation* parameter may contain no information or be absent and the *fileReference* and *fileExpiryDate* shall be present. The file that captures the test result data set shall contain VSE attributes and other 3GPP attributes such as *testerObjectClass*, *testOutcome*, etc.

The use of the *additionalInformation* parameter or a file to capture the test result data set is specified by the class specification of the TO.

In case the TO uses *additionalInformation* (and not a file) to capture the test result data set, that TO may emit this notification to transfer intermediate (non-final) test results. In this kind of notifications, the *testOutcome* parameter shall be absent. The TO should emit at least one more notification regarding the subject test invocation in the future. The last notification pertaining to a particular test invocation shall be indicated by including the *testOutcome* parameter in the notification.

In the case the TO uses a file to capture the test result data set, that TO shall not issue any notifications to transfer intermediate test results. The TO may capture the non-final test results in the file used to capture the final test result data set.

The events triggering the emission of test result notifications depend on the specific test. They shall be specified by the TO that is actually instantiated, i.e. either by the test category specific TO or the VSE TO. Some generic triggering events are included in this specification. It is expected that vendors specify more triggering events.

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M, Y	TesterObject.objectClass	Notification header - see [1]. It shall carry the TO class name.
objectInstance	M, Y	TesterObject.objectInstance	Notification header - see [1]. It shall carry the DN of the TO..
notificationId	O, N	--	Notification header - see [1].
eventTime	M, Y	--	Notification header - see [1].
systemDN	C, Y	--	Notification header - see [1].
notificationType	M, Y	"notifyTestResults"	Notification header - see [1].
testInvocationId	O, N	TestInvocation.testInvocationId	
testInvocationInitiator	C, Y	TesterObject.testInvocationInitiator	
testOutcome	O, N (see note)	TesterObject.testOutcome	It shall be included only in the last notification emitted by a TO. In this way the TO indicates that it is sending no more notifications.
mORT	O, N	TesterObject.theMORT	It identifies the object instance of the MORT that was subject to the test.
proposedRepairActions	O, N	TesterObject.proposedRepairActions	
additionalInformation	O, N (see note)	TesterObject.additionalInformation	It allows the inclusion of any additional information in the notification. As such, it may carry a test result data set. The exact semantics of this parameter is outside the scope of this specification. This parameter may contain no information or be absent, if the test results are captured in a file. It may be present if the test results are not captured in a file.
fileReference	M, N (see note)	TesterObject.fileReference	It shall contain no information or be absent if there is no test result captured in a file. It shall contain information if the test results are captured in a file.
fileExpiryDate	M, N (see note)	TesterObject.fileExpiryDate	It shall contain no information or be absent if fileReference carries no information or absent. Otherwise, it shall contain a valid future date and time.
NOTE: As for the correct interpretation of this qualifier refer to the comment column.			

6.5.1.2 Triggering Events for the Test

For the test the events triggering the emission of test result notifications are:

- Termination of the test execution.

The test may be terminated explicitly by a test termination request. The events triggering an implicit termination are:

- Fulfilment of the conditions for a successful termination of the test.

- Fulfilment of the conditions for a premature termination of the test.

- Occurrence of an error situation.

6.5.1.2.1 From-State

testTerminateRequestReceived OR testCompleted OR prematureTermination OR testTimedOut OR errorSituationOccured.

Assertion Name	Definition
testTerminateReque stReceived	The object with the ability to receive and react upon test requests has received a test termination request (see note).
testCompleted	The predefined conditions for a successful completion of the test are fulfilled (see note).
prematureTerminati on	The predefined conditions for a premature termination of the test are fulfilled (see note).
errorSituationOccu red	An error situation has occurred during the test execution and the tester object has aborted the test invocation (see note).
NOTE: The conditions to satisfy this trigger are related to the VSE TO definition and therefore, their specifications are outside the scope of 3GPP.	

6.5.1.2.2 To-State

testTerminated

Assertion Name	Definition
testTerminated	The test has been terminated successfully.

Annex A (informative): Network Performance Measurement Procedure

There are four procedures in network performance measurement: initiation, termination, monitor and test result report.

In the initiation, NM sends an initiation request in which there are some configuration parameters to EM with operation *initialTests*. The EM will store the parameters and send some configuration parameters to the source NE and destination NE (or the loopback NE) to initiate the network performance measurement. For the IP network, the network elements will start the test with the protocols defined in IETF [12] [13]. The following figure illustrates the test initialisation procedure.
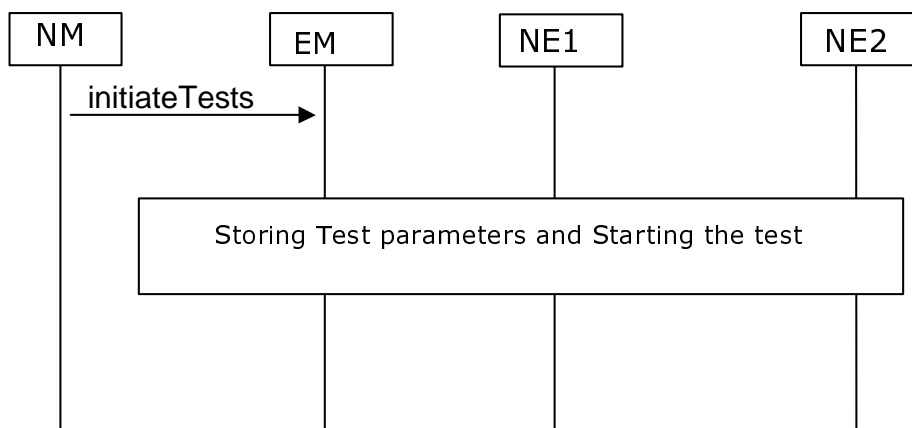

Figure A.1 Initiation procedure for Network Performance Measurement

When the NM terminates the test, NM sends a termination request to EM with operation *terminateTests*. The EM will save and send some parameters to the source NE and destination NE (or the loopback NE) to terminate the network performance measurement. The following figure illustrates the test termination procedure.
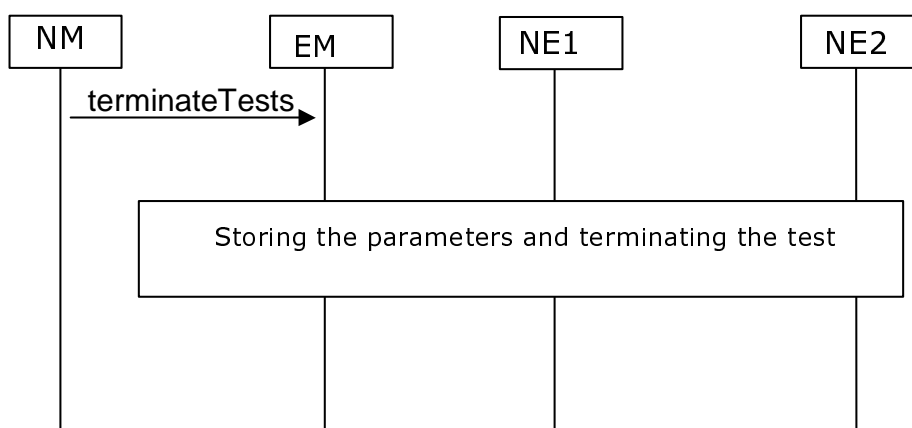

Figure A.2 Termination procedure for Network Performance Measurement

When the NM monitors the test, NM sends a monitor request to EM with operation *monitorTests*. If it is needed, the EM will send request to the NE(s) to retrieve information before it send back the output parameters. Then the EM will send back the output parameters to the NM. The following figure illustrates the test monitor procedure.
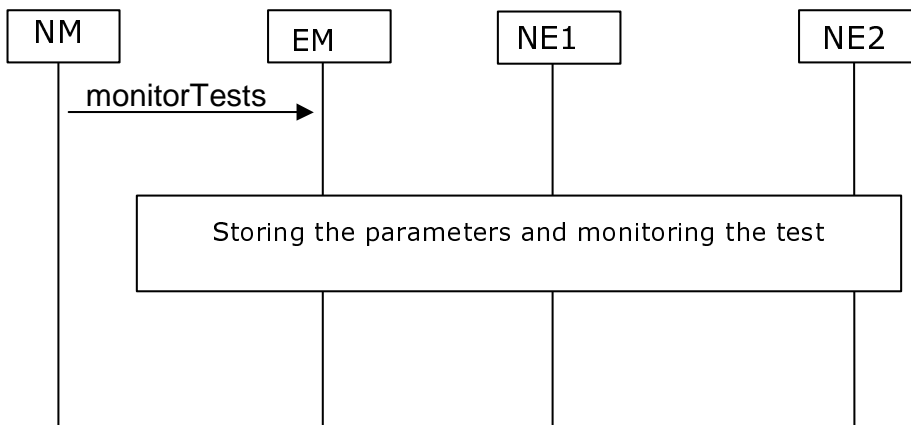
Figure A.3 Monitor procedure for Network Performance Measurement

After the test is terminated, the test result is recorded in a test result file in EM. Then the EM sends a notification *notifyTestResults* to the NM, in which there is some information for the NM to retrieve the file. The following figure illustrates the test monitor procedure.
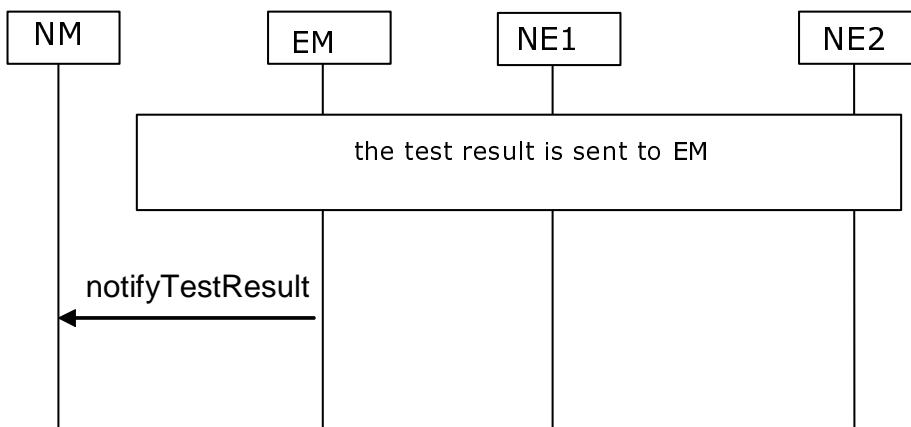

Figure A.4 Test result report procedure for Network Performance Measurement

Annex B (informative): Change history

Change history								
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Cat	Old	New
Jun 2002	SA_16	SP-020328	--	--	Submitted to TSG SA #16 for Information	--	1.0.0	
Sep 2002	SA_17	SP-020457	--	--	Submitted to TSG SA #17 for Approval	--	2.0.0	5.0.0
Dec 2002	--	--	--	--	Cosmetics	--	5.0.0	5.0.1
Jun 2004	SA_24	SP-040243	0001	--	Add missing parameter to the operation initiateTests	F	5.0.1	5.1.0
Sep 2004	SA_25	SP-040541	--	--	Automatic upgrade to Rel- 6 (no CR) as per request in SP-040541 SA5_presentation_SA_25.ppt (slide 17)	--	5.1.0	6.0.0
Jun 2005	SA_28	SP-050289	0002	--	Apply Generic System Context	F	6.0.0	6.1.0
Mar 2006	SA_31	SP-060089	0003	--	Update mapping table to use defined filter qualifier	F	6.1.0	6.2.0
Jun 2007	SA_36	--	--	--	Automatic upgrade to Rel-7 (no CR) at freeze of Rel-7. Deleted reference to CMIP SS, discontinued from R7 onwards.	--	6.2.0	7.0.0
Sep 2007	SA_37	SP-070606	0005	--	Correct granularity of maxTestingStateDuration	A	7.0.0	7.1.0
Mar 2008	SP-39	SP-080204	0008	1	Remove toBeMonitoredAttributes	F	7.1.0	7.2.0
Jun 2008	SP-40	SP-080329	0009	--	Add new Information Objects for connection and loopback test categories - Align with 32.321	B	7.2.0	8.0.0
Dec 2009	-	-	-	-	Update to Rel-9 version (MCC)	-	8.0.0	9.0.0
Mar 2011	-	-	-	-	Update to Rel-10 version (MCC)	-	9.0.0	10.0.0
2012-09	-	-	-	-	-	Update to Rel-11 version (MCC)	10.0.0	**11.0.0**
2014-10	-	-	-	-	-	Update to Rel-12 version (MCC)	11.0.0	**12.0.0**

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2020-07	SA#88e	SP-200415	0014	-	A	draft-ietf-ippm-twamp has been published as RFC5357	12.1.0

History

Document history		
V12.0.0	October 2014	Publication
V12.1.0	August 2020	Publication