International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Series X

Supplement 5

(09/2008)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

ITU-T X.290-series – Supplement on interoperability testing framework and methodology

ITU-T X-series Recommendations – Supplement 5

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339

For further details, please refer to the list of ITU-T Recommendations.

Supplement 5 to ITU-T X-series Recommendations

ITU-T X.290-series – Supplement on interoperability testing framework and methodology

Summary

Supplement 5 to ITU-T X.290-series Recommendations defines the principles, methodology and architectures to serve as a foundation for interoperability testing and the development of interoperability test suites. It is applicable to interoperability testing of protocols and also to interoperability testing of all areas of software.

Source

Supplement 5 to ITU-T X-series Recommendations was agreed on 19 September 2008 by ITU-T Study Group 17 (2005-2008).

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this publication, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this publication is voluntary. However, the publication may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the publication is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the publication is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this publication may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the publication development process.

As of the date of approval of this publication, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this publication. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Introduction

The need for interoperability testing arises when it is required to demonstrate that two or more systems or system components are capable of communicating with each other. Conformance testing (CT) checks whether an implementation is implemented correctly with respect to the requirements stated in the relevant standard or specification. A framework and methodology for conformance testing is specified in the ITU-T X.290-series Recommendations. By itself, it has two important limitations in ensuring interoperation of communicating systems. First, specifications may contain non-interoperable components such as optional features. Second, complete conformance testing may be impractical due to the large number of tests that may be required to cover all the requirements. Therefore, interoperability testing (IOPT) on multiple implementations is needed to complement conformance testing.

This supplement is intended to assist ITU-T Study Groups responsible for developing protocol specifications, conformance test suites and interoperability test suites by defining a testing methodology that includes a framework for test purposes, a test architecture and test methods for use in interoperability testing. It is also intended as a guide to the developers of interoperability test suites, test tool developers and test operators, with the objective of achieving a common framework for testing. Specification writers would understand non-interoperation phenomena and their causes better so that they can be more informed and prepared to write interoperable specifications. System developers would be helped in developing interoperable systems and services. Interoperability test suites developers would know how to write practically complete interoperability test suites (i.e., they would know what needs to be tested and would also know the gap between what should be tested and what can be tested). Test operators would get help in understanding the causes of interoperability problems and in finding out how to fix problems.

This supplement is complementary to, and may be used in conjunction with, testing methodologies and test specification languages defined in related Recommendations such as:

- ITU-T X.290 through ITU-T X.296, and
- ITU-T Z.161 through ITU-T Z.163,

where the X.290-X.296 series addresses conformance and provides the methodology and framework for conformance, and the Z.161-Z.163 series defines a general test scenarios description language for conformance testing and interoperability testing. A particularly related document in terms of its technical contents is Supplement 4 to ITU-T X-series Recommendations: *ITU-T X.290-series: Supplement on generic approach to interoperability testing*, which provides general guidance on the specification and execution of interoperability tests for communication systems in next generation networks (NGN). It is meant to be applied to communication equipment interoperability testing and in many places gives specific guidance on how test suites can be written and interoperability testing can be carried out. On the other hand, this Supplement 5 to the ITU-T X.290-series of Recommendations targets at more foundational principles and issues that underlie specific interoperability testing approaches and on the basis of which interoperability testing approaches can be carried out and assessed. In addition, Recommendations concerning ITU-T formal languages such as ASN.1, MSC, SDL and UML can be used in relation with this supplement, for test data description, system behaviours and test scenarios descriptions, communication system modelling, and software modelling, respectively.

Supplement 5 to ITU-T X-series Recommendations

ITU-T X.290-series – Supplement on interoperability testing framework and methodology

1 Scope

This supplement provides a generic framework and methodology for interoperability testing of interconnected communication systems. The framework is complementary to, and may be used in conjunction with, testing methodologies and test specification languages defined in related Recommendations. This supplement is also intended to assist study groups responsible for developing protocol specifications and conformance test suites.

This methodology is applicable to interoperability testing of protocols and also to interoperability testing of all areas of software. It is also intended as a guide to the developers of interoperability test suites, test tool developers and test operators, with the objective of achieving a common framework for testing.

2 References

[ITU-T X.290] Recommendation ITU-T X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts.*

[ITU-T X.291] Recommendation ITU-T X.291 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Abstract test suite specification.*

[ITU-T X.292] Recommendation ITU-T X.292 (2002), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – The Tree and Tabular Combined Notation (TTCN).*

[ITU-T X.293] Recommendation ITU-T X.293 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Test realization.*

[ITU-T X.294] Recommendation ITU-T X.294 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Requirements on test laboratories and clients for the conformance assessment process.*

[ITU-T X.295] Recommendation ITU-T X.295 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Protocol profile test specification.*

[ITU-T X.296] Recommendation ITU-T X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements.*

3 Definitions

This supplement defines the following terms:

3.1 application programming interface (API): Allows a software application to make use of another piece of software through well-defined software functions and procedures with fully specified parameters.

3.2 conformance testing (CT): A service, offered to clients by a test laboratory, to perform the conformance assessment process for one or more protocols, with a choice of test methods sufficient to make the service applicable to all real systems that claim to implement the specified protocols.

3.3 interoperability (IOP): The degree to which a set of entities can collaborate using a predetermined set of functions.

3.4 implementation under test (IUT): The subdivision of a system under test that is to be tested, such as, the specific layer(s) of protocol within a set of layers of protocol.

3.5 interoperability of communication systems: Ability of systems to interoperate using the relevant communication standard(s).

3.6 interoperability testing (IOPT): Activity of demonstrating that two or more entities, such as systems, equipment and software components, interoperate.

3.7 IOPT test case (IOPT-TC): A test case that tests interoperability of a set of entities.

3.8 IOPT test report (IOPT-TR): A document produced at the end of an interoperability testing carried out using a particular test suite. It provides the details of the test results by listing the test cases of the test suite that were executed, their execution scenarios and the verdicts assigned as the results of execution.

3.9 IOPT test suite (IOPT-TS): A set of interoperability test cases.

3.10 protocol data unit (PDU): An atomic unit of data specified in an (N)-layer protocol, consisting of (N)-protocol information and possibly (N)-user data.

3.11 protocol implementation conformance statement (PICS): A statement made by the supplier of an implementation or system, stating which capabilities have been implemented for a given protocol.

3.12 service access point (SAP): A physical interface of a component through which services provided at the interface are utilized.

3.13 test campaign: The process of executing an interoperability test suite against two IUTs.

3.14 test purpose (TP): A concise statement of the purpose of an interoperability test to be conducted on a pair of PUTs (or IUTs), in terms of the functionality of the relevant OSI application (profile or protocol), that is also independent of the PUTs (or IUTs) themselves.

3.15 test report (TR): A document which uniquely and fully identifies the conditions of an interoperability test campaign and the identity of the tests executed and presents the verdicts obtained for each test.

3.16 test specification: An amplification of a test purpose defining the test in more detail in terms of the OSI operations to be performed. The specification includes information on initial conditions for the test, expected behaviour and final conditions, together with criteria for judging Pass, Fail and Inconclusive test verdicts.

4 Abbreviations and acronyms

This supplement uses the following abbreviations and acronyms:

API	Application Programming Interface
ATM	Asynchronous Transfer Mode
CT	Conformance Testing
CT-TS	CT Test Suite
CTMF	Conformance Testing Methodology and Framework

FSM	Finite State Machine
ICS	Implementation Conformance Statement
IOP	Interoperability
IOPT	Interoperability Testing
IOPT-TC	IOPT Test Case
IOPT-TR	IOPT Test Report
IOPT-TS	IOPT Test Suite
IUT	Implementation Under Test
IXIT	Implementation eXtra Information for Testing
LT	Lower Tester
MOT	Means of Testing
PCO	Point of Control and Observation
PO	Point of Observation
PDU	Protocol Data Unit
PUT	Protocol Under Test
SAP	Service Access Point
TC	Test Campaign
TCP	Test Coordination Procedure
TMP	Test Management Protocol
TP	Test Purpose
TR	Test Report
UT	Upper Tester

5 General

5.1 Interoperability

When two or more entities[1] interact by exchanging or sharing information to perform a certain task as in the case of distributed applications and communication protocols, the capability to do so is called *interoperability* (IOP).[2] Interoperability is considered the essential aspect of correctness of collaborating entities. Here, 1) involvement of more than one entity and 2) the entities together behaving as expected, are the two key characteristics of (correct) interoperation. Thus, it can be said that two or more entities *interoperate* if they behave together as expected.

---

[1] The nature of entity: an entity may be a process, an abstract data type instance, a communication channel, which is a subject of a computation, storage and communication. Interface, in contrast, is the way through which two entities meet with each other. Interface, in itself, can be considered to be the "external shape. "If an entity is contents, its interface with the external world would be the various forms it can take.

[2] Interoperation can be contrasted with *cooperation* in that although they both represent a relation between entities that try to achieve certain goals, in cooperation the entities can be independent of each other, i.e., they might not exchange or share information.

These notions of interoperation and entity allow you to view various kinds of interacting behaviour as special kinds of interoperation and anything you decide to view as a whole can be considered as an entity.[3] Within the context of distributed applications and communication protocols, an entity can be, for example, a peer of the peer-to-peer type, a master or a slave of the master-slave type, or a client or a server of the client-server style distributed applications or protocols. If one views the entity as a system consisting of interacting subsystems, each sub-component is an entity.[4]

5.2 Interoperability testing

Interoperability can be seen at two levels: one at the specification level and the other at the implementation level. Here we only consider the latter. The activity of verifying interoperability of specification is called *interoperability validation*, and the activity of verifying or measuring interoperability between implementations is called *interoperability testing* (IOPT).[5] Interoperability validation checks whether the specification itself is correct so that the implementations based on the validated specification would be able to achieve interoperability.

5.3 Overview of the interoperability testing framework and methodology

The interoperability testing framework defines relevant fundamental concepts, assumptions and principles. The interoperability testing methodology refers to test methods and techniques. Once the framework and the methodology have been defined, practitioners should be able to apply them in the development of related standards and the testing activity. Interoperability test operators should be able to select the appropriate test architectures and test suites, execute correct test procedures and report the test results.

An interoperability test method as defined here consists of two activities. One is the method for test suite development and the other is the method for test execution. For test suite development, the test suite developer needs to consider test coverage, test architecture, test suite structure, test purposes, test scenarios, etc. For test execution, the test operator needs to consider test coverage, test case selection, static interoperability review, dynamic interoperability testing, test verdict assignment, test report, etc. The relevant steps of these activities are carried out in a specific order. Although these processes are intended to be adaptable to automation, some aspects of the processes are left up to the user as a choice within the general framework.

6 Interoperability testing framework

6.1 Interface, entity, association

An implementation under test can exist in isolation or in association with one or more interoperating entities. The point of interaction is called the *interface*.

Systems under test may consist of entities that are interrelated and may interact with each other. Such systems may provide services or functionalities to other systems through their interactions. An entity can be a subsystem, which can be decomposed into constituent smaller systems, or it may be an atomic entity, which does not have a further interaction substructure. All interactions take place through a well-defined interface.

---

[3] An entity can be a computing entity, a storage entity or a communication channel entity, depending on its functionality. An entity can be classified as active or passive, depending on the way it acts.

[4] The term *entity* in this Supplement can refer to a real world entity or an entity of a software system that is used to solve real world problems. In the latter case, the entity can be a computing entity, a storage entity, a communication channel entity or a combination of them.

[5] Verifying interoperability between specifications is called *validation*.

In order to define interoperability more rigorously, the basic modelling constructs are introduced in Figure 6-1 which depicts software systems from the structural point of view. These can be used to describe: 1) the models; 2) the structure or architecture of software systems; and 3) the architecture of software test systems.

Figure 6-1 – Fundamental constructs

The system is viewed as a "black box", where the internal operation may be either considered proprietary in nature or desirable to be ignored for testing purposes. The system is denoted by a box with rounded corners. The system may consist of subsystems.

Each system or subsystem may have zero or more interfaces. An interface may be viewed as a "window" through which an entity communicates with the outside world. The interface is binary so that exactly two entities can share one interface. Entities with no interfaces do not contribute to interoperability.

Each incarnation or instance of an interface has its *type* (interface type) and its *role*. The role can be either the initiator role or the responder role. For notational purposes, a role can be thought of as a "polarity" and designated with the "+" or "–" superscripts. In order for an interface to work, the role of the two participating entities should be of opposite polarity. A participating entity may sometimes behave as the initiator and sometimes as the responder. Figure 6-2 demonstrates interfaces identified alternatively only by name, by name and type, and by name, type and role.

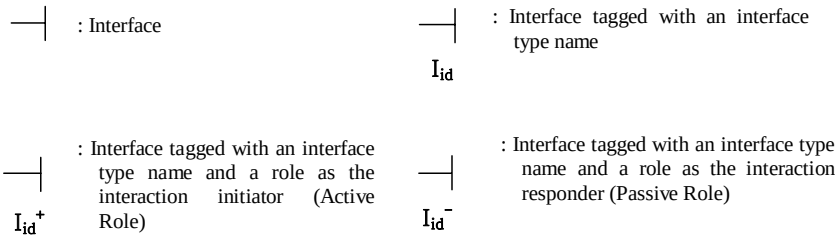


Figure 6-2 – Interface notations

Association indicates that two entities are not isolated from each other. If A and B are associated, either A knows B (hence A could be the user of B), B knows A or A and B know each other. Association is not a medium for information transfer. Therefore, entities A and B can be associated through the (information transfer) medium, entity M, as shown in Figure 6-3. In many cases, one is interested in knowing that A and B are associated but not interested in how such association is physically realized through a specific medium. Figure 6-4 shows a configuration where the medium M is abstracted.

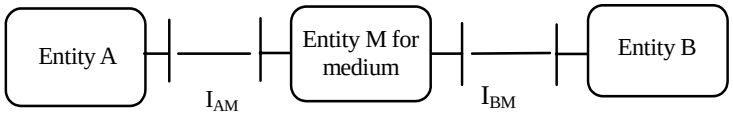


Figure 6-3 – Structure in which entities A and B are associated through entity M

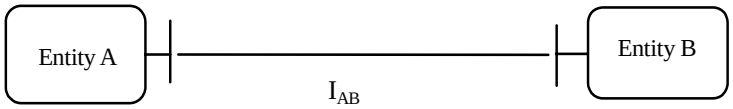


Figure 6-4 – Structure in which entities A and B are associated

6.2 Interoperation architecture

The following assumptions are made so as to focus on the essential aspects of interoperability testing for developing the interoperability testing framework and methodology:

1) The number of entities considered for interoperability testing is two.

2) The number of interfaces of an entity is at most two.

However, when the framework and methodology is applied, these assumptions can be removed and the framework and methodology can be generalized without difficulty for any number of interoperating entities.

According to the definition of interoperability in clause 5.1, the simplest kind of interaction behaviour can occur in the situation depicted in Figure 6-5. In this example, entity A and entity B can interact with each other through their only interface (I_{AB}).

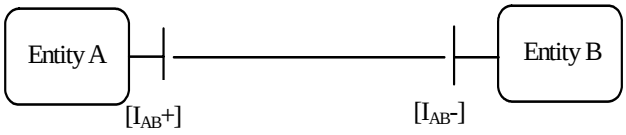


Figure 6-5 – Two interacting entities

A more complicated case arises when either entity A or entity B has more than one interface. Figure 6-6 depicts such a case. In Figure 6-6, entity A has two interfaces and can communicate through interface I_{AB} and through interface I_A. The configuration in Figure 6-6 is useful for applications using client-server architecture. Entity B may be a server and entity A a client. Interface I_A may be a graphic interface for the human user.

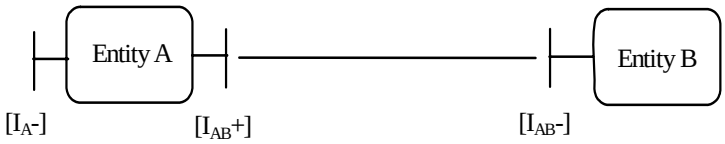


Figure 6-6 – Client-server architecture example

In Figure 6-7, entity B has an additional interface I_B. The architecture in Figure 6-7 can arise when entity A and entity B are peers of a communication protocol (or a communication protocols stacks). In such a case, the communication protocol implemented by entity A and entity B can be thought of as providing communication services to the users of entity A and entity B. The interfaces I_A and I_B could be defined as service access points (SAPs). I_A and I_B may be two instances of one type of SAP. If entity A and entity B are related as in a client-server architecture or a master-slave architecture, then I_A and I_B would likely be different types of SAPs.

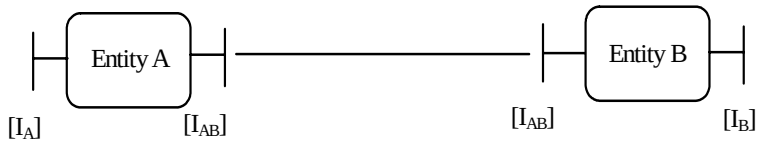


Figure 6-7 – Peer-to-peer architecture example

Figure 6-8 depicts a master-slave architecture. The unidirectional association is indicated with the arrow. The interpretation is that A knows B but B does not know A. Such a case arises when B is controlled by A. Entity A is then viewed as master and entity B as slave.[6]

Figure 6-8 – Master-slave architecture example

6.3 Interoperability test architecture

A tester is a component of the complete test system. There are two types of testers: 1) active testers and 2) passive testers. The term "tester" will indicate the active tester and the term "monitor" will indicate the passive tester. They are different in that the monitor (passive tester) can only observe interactions, whereas the tester (active tester) can interact by observing as well as by intercepting, inserting and modifying.

Each tester is able to make decisions about the correctness of the behaviour of the target system from its point of control and observation. The entity that is the target of testing is called implementation under test (IUT).

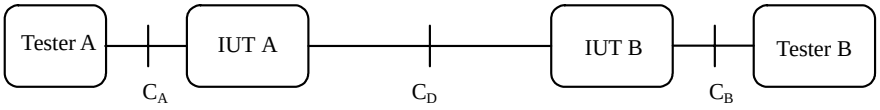
Systems exist in a wide variety of architectures and vary in the ways in which their behaviour can be controlled and observed during testing. In many cases, more than one test architecture can be used for interoperability testing. Under the assumptions of clause 6.2, various IOP test architectures are possible. For example, for the system with the architecture in Figure 6-7, the three generic interoperability-testing architectures depicted in Figure 6-9 can be considered. In Figure 6-9, a) depicts a test architecture with two testers, b) depicts a test architecture with two testers and one monitor and c) depicts a test architecture with three testers.

---

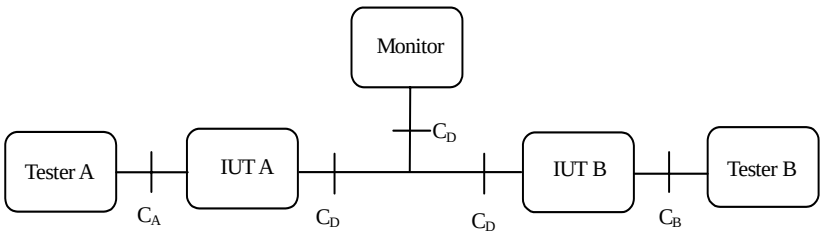
[6] In real life protocols, however, the slave may report, for example, an abnormal situation to the master; an exception to the normal communication direction. To depict such a situation precisely we have to use Figure 6-6.

The three architectures allow for different capabilities on controlling and observing interactions between the two IUTs. For example, in architecture I neither control nor observation over the interaction between IUTs is possible, whereas in architecture II only monitoring of the interactions is possible and in architecture III both control and observation over the interactions are possible. In architecture II, the monitor observes the interaction without affecting it. Thus, architecture II is more powerful than architecture I. In architecture III, Tester D can interfere with the interaction between two IUTs and can perform a wide range of test scenarios between the two IUTs. Architecture III is the most powerful and provides the superset of the capabilities of the other two.

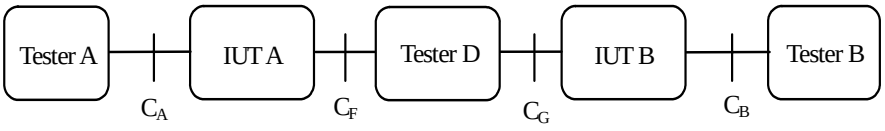
By focusing on the target system architecture, and considering the exhaustive set of relevant interfaces, we can derive several candidate test architectures. One or more test architectures are chosen from the candidate architectures choosing the most appropriate in terms of test coverage and test cost. In general, to minimize test cost, the number of tested interfaces is minimized.



a) IOP test architecture I



b) IOP test architecture II



c) IOP test architecture III

Figure 6-9 – Three IOP test architectures for peer-to-peer system testing[7]

6.4 Interoperability test case

An interoperability test case consists of a test purpose and a test scenario. A test purpose is a statement that specifies what the test case is to verify, and a test scenario is a sequence of interactions or test events used to control and observe the system during testing. The test scenario of

[7] There is no restriction on channels C_A and C_B. They can be physical links, SAPs, APIs or whatever as long as they provide the means to give inputs to the IUTs and take outputs from the IUTs. Thus, inputs from Testers A and B to the IUTs may be in various forms: e.g., messages, primitives. The other channels, C_D, C_F and C_G, are physical links, wired or wireless. Note that in b) we need to make explicit the communication entity between IUT A and IUT B, which was abstract in a).

an interoperability test case consists of three main parts: test preamble, test body and test postamble.[8]

Preamble is the part that puts the system into a stable state in which the test body can start. *Test body* is the part of the test case that tests the behaviour and assigns verdicts based on operational and transfer errors. The test body ends in a stable state. *Postamble*, which may be empty, verifies whether the system reached the expected stable state after the test body is completed.

6.5 Interoperability test suite

A collection of related test cases for interoperability testing of a system is called an *interoperability test suite* for the system. In order to derive test cases, test architectures should be first chosen. Multiple test architectures may be used for a single test suite. Therefore, generally a test suite can be considered to have the structure depicted in Figure 6-10.

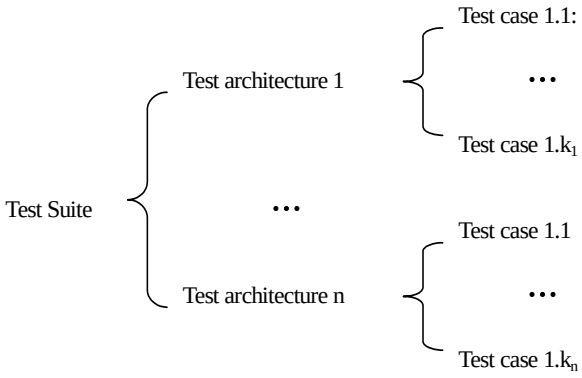


Figure 6-10 – Test suite structure

6.6 Test execution and verdict assignment

When a test case is executed, various testers may run in parallel depending on the interfaces exposed by the target system. During the execution, test components may make partial evaluation about the correctness of the target system. A preliminary evaluation result for the test case will be based on these partial results. At the end of a test case execution, the results are summarized as a unique test case verdict.

A preliminary verdict can take the value of `none`, `pass`, `fail` or `inconclusive` and the final verdict can take the value of `pass`, `fail` or `inconclusive`. Table 6-1 shows the current verdict value when combined with a new verdict for recently observed behaviour as test execution progresses.

[8] In conformance testing, the conformance test suite is developed in such a way that for the given specification with an FSM structure, absence of operation errors (or correctness of input/output behaviour), and absence of transfer errors (or correctness of the state reached after the transition) are examined with respect to each transition of the FSM. For this purpose, a conformance test case also consists of preamble, test body and postamble.

Table 6-1 – Verdict computation

Current verdict value	Verdict value on most recently observed behaviour		
	pass	inconc	fail
none	pass	inconc	fail
pass	pass	inconc	fail
inconc	inconc	inconc	fail
fail	fail	fail	fail

With increased complexity of the test architecture, computation of n verdicts is required in an n-level hierarchy of testers. Table 6-1 shows the 2-level verdict composition. According to Table 6-1, verdict values are ordered as:

none < pass < inconc < fail

If we let v denote the new verdict, v is calculated as

$$v = \text{lub}\{v_0, v_1, v_2, .. , v_n\}$$

where lub is the least upper bound, v_0 is the old verdict and v_i, $1 \leq i \leq n$, is a set of recently obtained verdicts.

7 Interoperability testing methodology

7.1 Interoperability test suite development

The definition of interoperation stated at the beginning of clause 5 is an *open* definition in the sense that it addresses the "expected behaviour" but does not state what the expected behaviour is. The expected behaviour differs from one system to another.[9] Interoperability requirements would be given explicitly and/or implicitly in specifications by specification writers. For the interoperability of distributed system and communication protocols, such requirements are usually given in more than one document. All requirements must be inferred or derived from the relevant specifications. Therefore, specification writers should be careful in writing specifications so as not to introduce non-interoperability into specifications through, for example, interoperability validation at the specification level.

Developers can develop test suites based on software artefacts at the various stages of software development such as system specification, design or implementation code. However, this documents deals only with the development of interoperability test suites based on standard requirements specification.

9 Still certain common interaction behaviour may be expected among a class of collaborating systems.

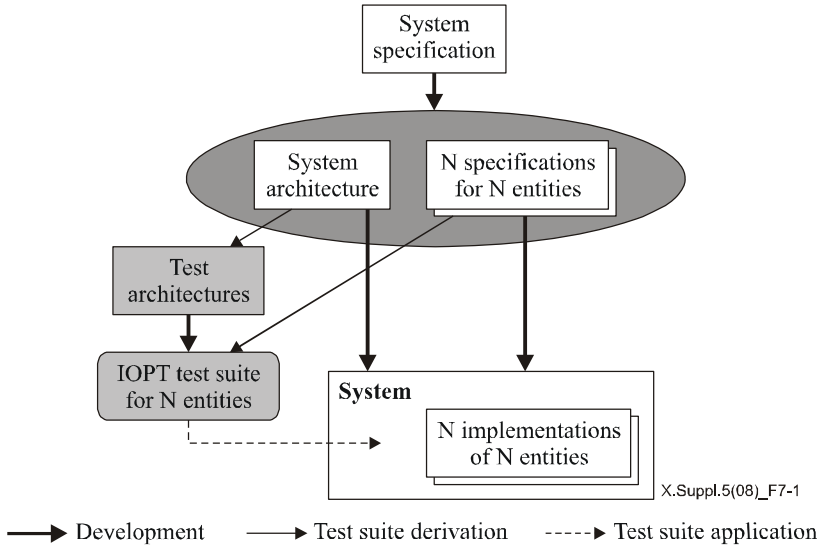


Figure 7-1 – Interoperability test derivation

In Figure 7-1, the test derivation method selects one or more test architectures from the given system architecture. Based on these selected architectures, the IOPT-TS representing N specifications of N entities is derived. This test suite is applied for testing the interoperability between the N entities that are represented in the given system architecture.

7.2 Interoperability testing procedure

In the following interoperability test procedure, it is assumed that there already exists an interoperability test suite.

The interoperability testing procedure can be summarized as depicted in the flowchart shown in Figure 7-2. The test operator receives the ICS and IXIT for the IUTs as stated in clauses 5.6 and 6.2 of [ITU-T X.290]. Based on the ICSs and IXITs, a static interoperability review is performed. If the review results indicate that interoperability testing should not continue with the dynamic testing, the final report is prepared and the testing ends. Otherwise, using ICS and IXIT information, a number of interoperability test cases are selected and prepared for dynamic interoperability testing. These tests are chosen for the IUTs that claim interoperability according to the standard (as claimed in the ICS). The selection step is necessary because the interoperability test suite may contain test cases to test functionalities for which interoperability is not claimed. Interoperability test cases are executed and results are analysed. A final test report is prepared.

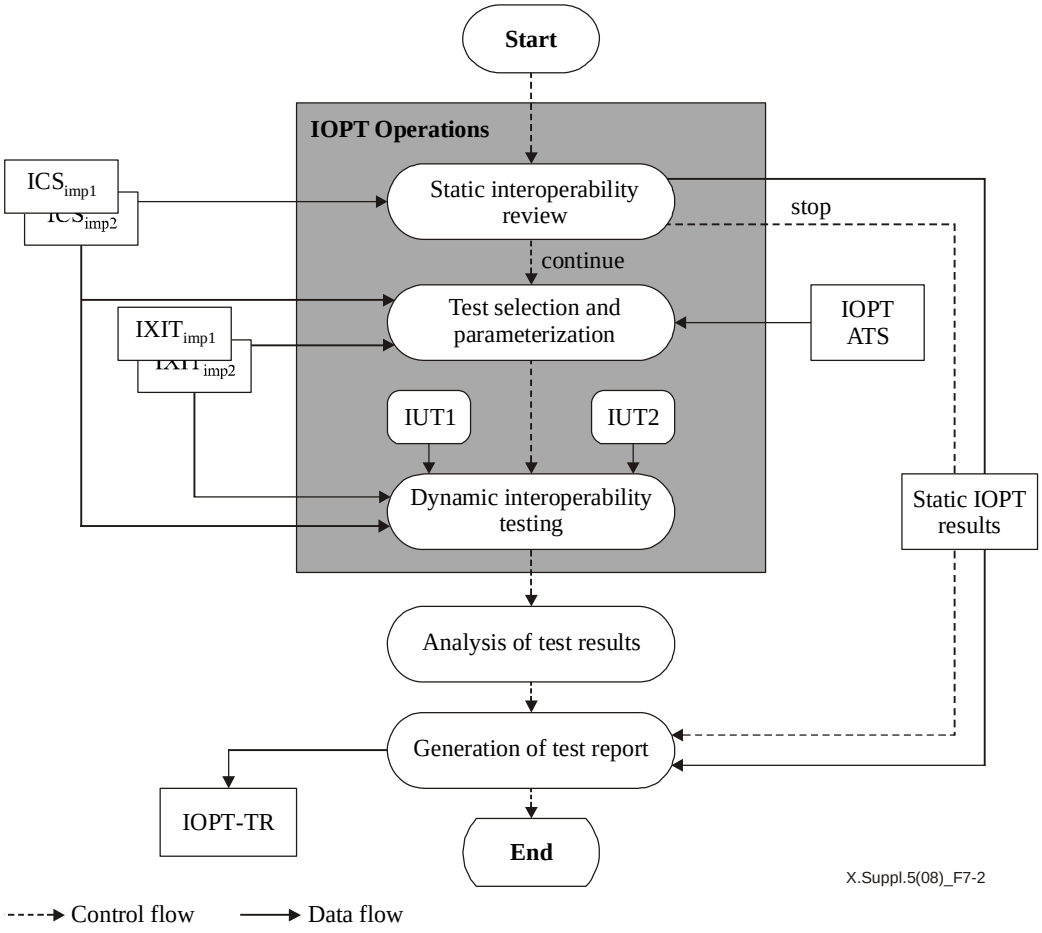


Figure 7-2 – Interoperability testing procedure

Bibliography

[b-ITU-T X.680] Recommendation ITU-T X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

[b-ITU-T X.681] Recommendation ITU-T X.681 (2002) | ISO/IEC 8824-2:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*

[b-ITU-T X.682] Recommendation ITU-T X.682 (2002) | ISO/IEC 8824-3:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*

[b-ITU-T X.683] Recommendation ITU-T X.683 (2002) | ISO/IEC 8824-4:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*

[b-ITU-T X.690] Recommendation ITU-T X.690 (2002) | ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*

[b-ITU-T X.691] Recommendation ITU-T X.691 (2002) | ISO/IEC 8825-2:2002, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).*

[b-ITU-T X.692] Recommendation ITU-T X.692 (2002) | ISO/IEC 8825-3:2002, *Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN).*

[b-ITU-T X.693] Recommendation ITU-T X.693 (2001) | ISO/IEC 8825-4:2002, *Information technology – ASN.1 encoding rules: XML Encoding Rules (XER).*

[b-ITU-T Z.100] Recommendation ITU-T Z.100 (2007), *Specification and Description Language (SDL).*

[b-ITU-T Z.105] Recommendation ITU-T Z.105 (2003), *SDL combined with ASN.1 modules (SDL/ASN.1).*

[b-ITU-T Z.109] Recommendation ITU-T Z.109 (2007), *SDL-2000 combined with UML*.

[b-ITU-T Z.110] Recommendation ITU-T Z.110 (2000), *Criteria for use of formal description techniques by ITU-T*.

[b-ITU-T Z.120] Recommendation ITU-T Z.120 (2004), *Message Sequence Chart (MSC).*

[b-ITU-T Z.150] Recommendation ITU-T Z.150 (2003), *User Requirements Notation (URN) – Language requirements and framework.*

[b-ITU-T Z.161] Recommendation ITU-T Z.161 (2007), *Testing and Test Control Notation version 3: TTCN-3 core language*.

[b-ITU-T Z.162] Recommendation ITU-T Z.162 (2007), *Testing and Test Control Notation version 3: TTCN-3 tabular presentation format.*

[b-ITU-T Z.163] Recommendation ITU-T Z.163 (2007), *Testing and Test Control Notation version 3: TTCN-3 graphical presentation format (GFT).*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2009