INTERNATIONAL TELECOMMUNICATION UNION

CCITT

X.291

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

DATA COMMUNICATION NETWORKS

**OPEN SYSTEMS INTERCONNECTION (OSI)
PROTOCOL SPECIFICATIONS,
CONFORMANCE TESTING**

OSI CONFORMANCE TESTING METHODOLOGY AND FRAMEWORK FOR PROTOCOL RECOMMENDATIONS FOR CCITT APPLICATIONS – ABSTRACT TEST SUITE SPECIFICATION

Recommendation X.291

Geneva, 1992

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT 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 Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation X.291 was prepared by Study Group VII and was approved under the Resolution No. 2 procedure on the 17th of January 1992.

CCITT NOTE

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

Recommendation X.291

OSI CONFORMANCE TESTING METHODOLOGY AND FRAMEWORK FOR PROTOCOL RECOMMENDATIONS FOR CCITT APPLICATIONS – ABSTRACT TEST SUITE SPECIFICATIONS[1)]

The CCITT,

considering

(a) that Recommendation X.200 defines the Reference Model of Open Systems Interconnection (OSI) for CCITT Applications;

(b) that the objective of OSI will not be completely achieved until systems can be tested to determine whether they conform to the relevant OSI protocol Recommendations;

(c) that standardized test suites should be developed for each OSI protocol Recommendation as a means to:

- obtain wide acceptance and confidence in conformance test results produced by different testers,
- provide confidence in the interoperability of equipments which passed the standardized conformance tests;

(d) the need for defining Recommendation to specify the framework and general principles for conformance testing;

(e) the need for defining Recommendation for the specification of conformance test suites and testing of protocol implementations,

unanimously declares the view

that the test methods and test suites shall be in accordance with this Recommendation.

CONTENTS

1) Recommendation X.291 and ISO 9646-2, Information technology – Open Systems Interconnection – Conformance Testing Methodology and Framework – Part 2: Abstract Test Suite Specification, are technically aligned.

0 Introduction

This Recommendation provides a common approach to the specification of OSI conformance test suites at a level which is independent of the means of executing those test suites (hereafter called “abstract test suites”). This level of abstraction is suitable for standardization and facilitates the comparison of results produced by different organizations which run the corresponding executable test suites.

Sections 6 and 7 recall that there are requirements on OSI protocol specifiers which have to be fulfilled before there can be an objective basis for the process of developing an abstract test suite. The need is expressed for consistent conformance clauses and for PICS proformas in CCITT Recommendations or International Standards which specify OSI protocol standards.

Sections 8 to 16 describe the process of developing an abstract test suite, including the design criteria to be used and guidance on its structure and coverage. The possible abstract test methods are defined and guidance is given to help the test suite specifier to decide which test method(s) to use in the production of a particular test suite. Requirements and guidance are given on the specification of abstract test cases. These include the subdivision of test cases into test steps and the assignment of test verdicts to test outcomes.

The test suite specifier is also required to provide information to the test realizers (e.g. limitations governing test case selection).

Finally, guidance and requirements are given on test suite maintenance.

This Recommendation is also published as ISO/IEC 9646-2: (1991).

1 Scope

1.1 This Recommendation specifies the requirements and gives guidance for the production of system-independent conformance test suites for one or more OSI CCITT Recommendations or International Standards. In particular, it is applicable to the production of all conformance testing standards for OSI and ISDN two-party protocols, including all draft versions of such conformance testing standards.

1.2 This Recommendation is applicable to the production of conformance test cases which check the conformance of an implementation to the relevant static and/or dynamic conformance requirements by controlling and observing protocol behaviour. The Abstract Test Methods included in this Recommendation are, in fact, capable of being used to specify any test case which can be expressed abstractly in terms of control and observation of Protocol Data Units and Abstract Service Primitives. Nevertheless, for some protocols, test cases may be needed which cannot be expressed in these terms. The specification of such test cases is outside the scope of this Recommendation, although the test cases may themselves need to be included in a conformance testing standard.

Note – For example, some static conformance requirements related to an Application service element may require testing techniques which are specific to that particular Application.

1.3 The following are outside the scope of this Recommendation:

a) the relationship between Abstract Test Suite specification and Formal Description Techniques;

b) testing by means of test methods which are specific to particular applications, protocols or systems, including the testing of non-protocol conformance requirements;

c) test methods that involve more than two end-systems communicating together.

Note – This Recommendation applies fully to some but not all Physical layer protocols. Nevertheless, many of the concepts apply to all protocols.

2 References

Rec. X.200 (1988) – *Reference Model of Open Systems Interconnection for CCITT Applications.* (See also ISO 7498: 1984)

Rec. X.210 (1988) – *Open Systems Interconnection Layer Service Definition Conventions.* (See also ISO/TR 8509: 1987)

Rec. X.209 (1988) – *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).* (See also ISO 8825: 1990)

Rec. X.290 (1992) – *OSI Conformance testing methodology and framework for Protocol Recommendations for CCITT Applications – General concepts.* (See also ISO/IEC 9646-1: 1991)

Rec. X.292 – *OSI Conformance Testing Methodology and Framework for Protocol Recommendations for CCITT Applications – The tree and tabular combined notation.* (See also ISO/IEC 9646-2[2])

3 Definitions

For the purposes of this Recommendation, all the definitions given in Recommendation X.290 apply.

4 Abbreviations

For the purposes of this Recommendation the abbreviations given in Recommendation X.290 apply. The following abbreviations also apply to this Recommendation:

CS: Coordinated single-layer (test method)

CSE: Coordinated single-layer embedded (test method)

DS: Distributed single-layer (test method)

DSE: Distributed single-layer embedded (test method)

FDT: Formal description technique

LS: Local single-layer (test method)

LSE: Local single-layer embedded (test method)

RS: Remote single-layer (test method)

RSE: Remote single-layer embedded (test method)

TCP: Test coordination procedures

TSS&TP: Test suite structure and test purposes

YL: Loop-back (test method)

YT: Transverse (test method)

5 Compliance

5.1 A CCITT Recommendation or International Standard which specifies a protocol in compliance with this Recommendation shall satisfy all the requirements stated in §§ 6 and 7 plus Annex A.

Note – Such compliance is a precondition for the protocol specification to be an effective basis for conformance testing of implementations.

5.2 An abstract test suite (ATS) specification which complies with this Recommendation:

a) shall be a conformance test suite;

b) shall be specified in a test notation standardized by CCITT or ISO/IEC;

c) shall satisfy all the requirements stated in §§ 9 to 14 inclusive;

[2] To be published.

d) shall be a CCITT Recommendation or International Standard or, in the absence of such a CCITT Recommendation or International Standard, shall be a publicly available document which is in the process of being standardized within CCITT or ISO/IEC, which has the highest standardization status available, and which has the status of at least a Committee Draft, Draft Recommendation or equivalent.

Note – ATSs outside the standardization process need to be submitted for international standardization before they can fully comply with this Recommendation, in order to ensure that they are subject to public scrutiny, correction and acceptance, internationally.

5.3 It is recommended that the test notation used be TTCN. If TTCN is used, the ATS shall comply with Recommendation X.292.

6 Conformance requirements in OSI CCITT Recommendations and International Standards

6.1 *Introduction*

The meaning of conformance in OSI is discussed in Recommendation X.290. It is necessary that there be an unambiguous and objective understanding of the conformance requirements of an OSI protocol or transfer syntax specification, as a prerequisite to the production of an ATS for that specification. Sections 6 and 7 state the requirements on protocol specifiers to ensure that there is such an understanding of the conformance requirements.

Additional guidance is given in Annex B.

6.2 *General requirements*

6.2.1 A clear distinction shall be made between static and dynamic conformance requirements. To avoid ambiguity, they should be stated separately from one another.

6.2.2 It shall be clear what conformance to the CCITT Recommendation or International Standard means, in the sense of what shall be done, what is permitted but not mandatory, and what shall not be implemented, in order to conform to it.

6.2.3 It shall always be decidable whether an instance of communication conforms dynamically or not.

For example, it should be possible to look at a record of PDU (protocol data unit) activity and decide whether or not it is valid with respect to the relevant CCITT Recommendation or International Standard.

6.3 *Conformance clauses*

6.3.1 Each CCITT Recommendation or International Standard which specifies an OSI protocol or transfer syntax shall include a conformance clause, which shall be expressed clearly and unambiguously.

6.3.2 Conformance clauses shall distinguish between the following categories of information:

a) references to clauses which state dynamic conformance requirements;

b) static conformance requirements concerning the protocol implementation;

c) static conformance requirements concerning multi-layer dependencies.

6.3.3 The requirement to produce a PICS, in compliance with the PICS proforma, shall be stated separately from the requirements on the protocol implementation itself.

6.3.4 The conformance clause should also include:

a) the requirement to be able to accept all correct sequences of PDUs received from peers, and respond with correct PDU sequences;

b) the requirement to be able to respond correctly to all incorrect sequences of PDUs received;

c) in connection oriented protocols, the option to support either the initiation of a connection or the acceptance of a connection, or both;

d) in connectionless protocols, the option to support the transmission of a PDU, the receipt of a PDU, or both.

7 Requirements on PICS proformas

7.1 The specific requirements to be met by suppliers in respect of each PICS they provide shall be stated in the relevant CCITT Recommendation or International Standard which specifies the protocol. The specification of these requirements shall include a PICS proforma. The PICS proforma shall be found in a separate part of the CCITT Recommendation or International Standard which specifies the protocol.

7.2 The PICS proforma shall be in the form of a questionnaire or checklist to be completed by the supplier or implementor of an implementation of the relevant OSI protocol.

7.3 The PICS proforma shall cover all optional and conditional functions, elements of procedure, parameters, options, PDUs, timers, multi-layer dependencies and other capabilities identified in the protocol specification.

There shall be a well-defined mapping (by references) from the PICS proforma to the static conformance requirements.

Annex A provides requirements and guidance on the construction of PICS proformas.

8 Abstract Test Suite production process leading to conformance testing standards

8.1 In order to present the requirements and general guidance for ATS specification, it is useful to assume a normal form of the process of ATS production leading to a conformance testing standard. This section describes the process in just such a normal form. ATS specifiers are not required to follow this normal form exactly, however they are recommended to use a similar process involving the same steps, possibly in a different order.

8.2 For the purposes of this Recommendation, the ATS production process is assumed to be as follows:

a) study the relevant specifications and PICS proformas to determine what the conformance requirements (including options) are which need to be tested (see § 9);

b) decide which test groups will be needed to achieve the appropriate coverage of the conformance requirements (see § 10.2);

c) optionally develop test group objectives: the common testing objectives of the elements of each test group (see § 10.3);

d) develop test purposes which reflect the test group objectives (if any) of the test groups in which they are contained, and which provide adequate coverage of the conformance requirements to be tested (see §§ 10.3 and 10.4);

e) optionally, specify generic test cases for each test purpose, using some appropriate test notation (see § 10.5);

f) choose the test method(s) for which the complete abstract test cases need to be specified, and decide what restrictions need to be placed on the capabilities of the lower tester and (if appropriate to the chosen test method(s)) the upper tester and test coordination procedures (see § 11);

g) apply a standardized test notation to specify the set of abstract test cases, including the test step structure to be used (see § 12);

h) specify the interrelationships:

   1) among the test cases,

   2) between the test cases and the PICS, and

3) as far as possible, between the test cases and the PIXIT, in order to determine the restrictions on the selection and parameterization of test cases for execution, and the restrictions, if any, on the orders in which they can be executed (see § 15);

i) consider the procedures for maintaining the ATS (see § 16).

8.3 It is also assumed that during the ATS production process an overall structure for the conformance testing standard(s) will be developed, with appropriate parts for:

a) the Test Suite Structure and Test Purposes (TSS&TP) (see § 10);

b) optionally, a generic test suite (see § 11);

c) one or more ATSs (see § 13) for one or more Abstract Test Methods (see § 12);

d) the specification (if applicable), of a test management protocol (TMP) (see § 14).

8.4 Sections 9 to 16 provide requirements and guidance which relate to each step in the above process.

9 Conformance requirements and PICS proforma

9.1 Before an ATS can be specified, the ATS specifier shall first determine what the conformance requirements are for the relevant protocol and/or transfer syntax specifications and what is stated in the PICS proforma concerning the implementation of those specification(s).

9.2 Sections 6 and 7 specify the requirements to be met by protocol specifiers as a prerequisite to the production of an ATS for a particular protocol.

9.3 If the static conformance requirements are not properly specified, the ATS specifier should contribute to the production of an amendment to or revision of the relevant CCITT Recommendation or International Standard to clarify the conformance requirements. Pending resolution of the problem, additional guidance for the ATS specifier is given in Annex C.

10 Test Suite Structure and Test Purposes

10.1 *Basic requirements*

10.1.1 The test suite structure and set of test purposes applicable to all ATSs to be specified for the same OSI protocol shall be specified in the relevant conformance testing standard, preferably in a separate part.

10.1.2 Each ATS shall comprise a number of test cases, each of which is designed to achieve one of the specified test purposes. The test cases may be grouped into test groups, if necessary nested. The structure shall be hierarchical; thus, an item at a lower level shall be completely contained within a higher level item. The structure need not, however, be strictly hierarchical: thus, any one test case may occur in more than one test suite or test group. Similar test groups may occur in more than one higher level test group or test suite.

10.1.3 The ATS specifier shall ensure that a subset of the test purposes of each ATS is concerned with capability testing, and another subset is concerned with behaviour testing. This need not lead to distinct test cases for behaviour and capability testing because it may be possible to use the same test steps for both a behaviour test purpose and for a capability test purpose. The ATS specifier shall provide an explanation of how the test purposes are derived from or relate to the protocol specification. The ATS specifier shall also provide a summary of the coverage achieved by the ATS.

10.2 *Specification of the test suite structure*

10.2.1 In order to ensure that the resulting ATS provides adequate coverage of the relevant conformance requirements, the test suite specifier is advised to design the test suite structure in terms of nested test groups in a top down manner.

There are many ways of structuring the same test suite into test groups; no one way is necessarily right and the best approach for one test suite may not be appropriate for another test suite. Nevertheless, the test suite specifier shall ensure that the test suite includes test cases for whichever of the following categories is relevant:

a) capability tests (for static conformance requirements);

b) behaviour tests of valid behaviour;

c) behaviour tests that investigate the reaction of the IUT to invalid test events; these may be subdivided into those concerned with syntactically invalid test events, semantically invalid test events, and inopportune test events, as relevant to the protocol concerned;

d) tests focusing on PDUs sent to the IUT;

e) tests focusing on PDUs received from the IUT;

f) tests focusing on interactions between PDUs sent and PDUs received;

g) tests related to each mandatory capability;

h) tests related to each optional capability;

i) tests related to each protocol phase;

j) variations in the test event occurring in a particular state;

k) timing and timer variations;

l) PDU encoding variations;

m) variations in values of individual parameters;

n) variations in combinations of parameter values.

This list is not exhaustive; additional categories might be needed to ensure adequate coverage of the relevant conformance requirements for a specific test suite. Furthermore, these categories overlap one another and it is the task of the test suite specifier to arrange them into an appropriate hierarchical structure.

10.2.2 The following structure is an example of a single-layer test suite, provided for guidance:

A Capability tests

A.1 Mandatory capabilities

A.2 Optional capabilities

B Behaviour tests: response to valid behaviour by peer implementation

B.1 Connection establishment phase (if relevant)

B.1.1 Focus on what is sent to the IUT

B.1.1.1 Test event variation in each state

B.1.1.2 Timing/timer variation

B.1.1.3 Encoding variation

B.1.1.4 Individual parameter value variation

B.1.1.5 Combination of parameter values

B.1.2 Focus on what is received from the IUT

– substructured as B.1.1

B.1.3 Focus on interactions

– substructured as B.1.1

B.2 Data transfer phase

– substructured as B.1

B.3 Connection release phase (if relevant)

– substructured as B.1

C Behaviour tests: response to syntactically or semantically invalid behaviour by peer implementation

C.1 Connection establishment phase (if relevant)

C.1.1 Focus on what is sent to the IUT

C.1.1.1 Test event variation in each state

C.1.1.2 Encoding variation of the invalid event

C.1.1.3 Individual invalid parameter value variation

C.1.1.4 Invalid parameter value combination variation

C.1.2 Focus on what the IUT is requested to send

C.1.2.1 Individual invalid parameter values

C.1.2.2 Invalid combinations of parameter values

C.2 Data transfer phase

– substructured as C.1

C.3 Connection release phase (if relevant)

– substructured as C.1

D Behaviour tests: response to inopportune events by peer implementation

D.1 Connection establishment phase (if relevant)

D.1.1 Focus on what is sent to the IUT

D.1.1.1 Test event variation in each state

D.1.1.2 Timing/timer variation

D.1.1.3 Special encoding variations

D.1.1.4 Major individual parameter value variations

D.1.1.5 Variation in major combination of parameter values

D.1.2 Focus on what is requested to be sent by the IUT

– substructured as D.1.1

D.2 Data transfer phase

– substructured as D.1

D.3 Connection release phase (if relevant)

– substructured as D.1

10.2.3 This test group structure does not cover basic interconnection tests. These may be provided as a list of selected capability and/or behaviour tests, but shall not involve any additional test purposes.

10.3 *Specification of the test purposes*

10.3.1 The test suite specifier shall create a set of test purposes, with each test purpose focused on a single conformance requirement of the relevant specification(s). It is suggested that test groups of related test purposes are identified first (as described in § 10.2) and that text defining the test group objective be produced for each test group. Within each test group, several more specific test objectives should be defined, to become either nested test group objectives or individual test purposes. By successive refinement of the test group objectives in this way, a structured set of test purposes may be produced.

The test purposes could be produced directly from studying those clauses in the relevant specification(s) which are appropriate to the test group concerned. For some test groups, the test purposes might be derivable directly from the protocol state table; for others, they might be derivable from the PDU encoding definitions or the descriptions of particular parameters, or from text which specifies the relevant conformance requirements.

This orderly construction technique helps to ensure the adequate coverage of the conformance requirements to be tested. It also avoids unnecessary duplication of text in the test purposes, because the full description of each test purpose does not have to be written explicitly, but can be assembled by tracing a path down through the nested structure of objectives.

Note – If the test suite specifier employs a formal description of the protocol(s) concerned, test purposes may be derived from that by means of some automated method. If an automated method is used, the same requirements apply. However, methods based on FDTs are outside the scope of this Recommendation. Nevertheless, if an FDT is to be used for this purpose, it is preferred that it be a standardized one.

10.3.2 In order to increase the efficiency of testing individual parameters on a single PDU, combined test purposes may be specified for a single abstract test case. Test purposes for invalid parameter values shall not be combined with other test purposes of valid or invalid values.

10.3.3 As part of the process of designing the TSS&TP, it is suggested that test purposes be identified initially for each specific parameter that is to be tested. As a second stage, combinations of individual parameter test purposes related to the same PDU may be specified. If this is done,

a) a combined test purpose shall be written, combining and referencing individual test purposes;

b) an indication shall be given that one abstract test case is to be produced for that combined test purpose, rather than a distinct test case for each of the individual test purposes that have been combined;

c) the individual test purposes shall remain in the set of test purposes, but shall reference the appropriate combined test purpose.

10.3.4 The result of defining and then combining particular test purposes is a specification of a test suite structure and a list of names of the test cases that shall apply to both the test purposes and to any ATS produced for those test purposes.

10.3.5 Whatever method is used to derive the test purposes, the test suite specifier should ensure, as far as possible, that they provide an adequate coverage of the conformance requirements of the relevant specification(s). There shall be at least one test purpose related to each distinct conformance requirement.

10.3.6 Test purposes should be specified not only for clearly testable conformance requirements, but also for conformance requirements that may be untestable using the test methods defined in this Recommendation.

Note – Test purposes for untestable requirements serve to inform protocol specifiers which conformance requirements are untestable, by indicating gaps in the standardized ATSs.

10.4 *Coverage*

It is possible to give guidance on the meaning of “adequate” coverage with reference to the test suite structure example in § 10.2. In order to express this, a shorthand notation will be used: the letter “x” will represent all appropriate values for the first digit in the test group identifier, and similarly “y” for the second digit, so that B.x.y.1 stands for B.1.1.1, B.1.2.1, B.1.3.1, B.2.1.1, B.2.2.1, B.2.3.1, B.3.1.1, B.3.2.1 and B.3.3.1.

With this notation, a minimum “adequate” coverage for the example given in § 10.2 is considered to be as follows:

a) for capability test groups (A.1, A.2)

   1) at least one test purpose per relevant capability,

   2) at least one test purpose per relevant PDU type and each major variation of each such type, using “normal” or default values for each parameter;

b) for test groups concerned with test event variation in each state (B.x.y.1, C.x.1.1, D.x.y.1), at least one test purpose per relevant state/event combination;

c) for test groups concerned with timers and timing (B.x.y.2, D.x.y.2), at least one test purpose concerned with the expiry of each defined timer;

d) for test groups concerned with encoding variations (B.x.y.3, C.x.1.2, D.x.y.3), at least one test purpose for each relevant kind of encoding variation per relevant PDU type;

e) for test groups concerned with valid individual parameter values (B.x.y.4, D.x.y.4)

   1) for each relevant integer parameter, test purposes concerned with the boundary values and one randomly selected mid-range value,

   2) for each relevant bitwise parameter, test purposes for as many values as practical, but not less than all the "normal" or common values,

   3) for other relevant parameters, at least one test purpose concerned with a value different from what is considered "normal" or default in other test groups;

Note – Tests for valid parameter values should focus on the relevant claims made in the PICS.

f) for test groups concerned with syntactically or semantically invalid individual parameter values (C.x.1.3, C.x.2.1)

   1) for each relevant integer parameter, test purposes concerned with invalid values adjacent to the allowed boundary values defined in the protocol specification, plus one other randomly selected invalid value,

   2) for each relevant bitwise parameter, test purposes for as many invalid values as practical,

   3) for all other relevant types of parameter, at least one test purpose per parameter;

Note – Tests for invalid parameter values should focus on values outside the range defined in the relevant protocol specification, rather than valid values outside the range claimed in the PICS.

g) for test groups concerned with combinations of parameter values (B.x.y.5, C.x.1.4, C.x.2.2, D.x.y.5)

   1) at least one test purpose for each important combination of specific values (e.g. boundary values),

   2) at least one test purpose per set of interrelated parameters to test a random combination of relevant values.

The test suite specifier shall not assume that the test realizer or test laboratory will perform any checking of test events against the values specified in the PICS other than that checking which is specified in the abstract test cases. Therefore, the test purposes and abstract test cases shall make explicit use of values given in the PICS whenever checking of valid parameter values is specified. The test suite shall include test cases to check for the support of parameter values that are allowed by the protocol specification and are within the ranges stated in the PICS. Such test cases shall make use of test suite parameters that contain the relevant PICS values. The test suite shall also include test cases to check for valid reactions to parameter values that are invalid with respect to the protocol specification. Parameter values that are valid with respect to the protocol specification(s) but outside the ranges stated in the PICS shall not be tested.

Note – The progression of the work on formal methods in conformance testing may provide analytical approaches to assess the appropriate coverage of an ATS, especially for the state/event variations, as in b) above. This Recommendation, however, does not recommend any particular analytical approach.

10.5 *TSS&TP compliance clause*

The TSS&TP part shall include a compliance clause concerning the development of test suites for that TSS&TP. That clause shall require, as a minimum, that a generic or ATS complying with the TSS&TP part:

a) consists of a set of test cases corresponding to the set or to a subset of the test purposes specified in the TSS&TP part;

b) uses a test suite structure which is an appropriate subset of the whole of the test suite structure specified in the TSS&TP part;

Note – The only subsetting of the test suite structure that should take place is the omission of test purposes that are untestable in the chosen Abstract Test Method. In particular, for embedded test method variants, this will be necessary due to the limitations imposed by the use of the protocol(s) above the one that is the focus of the test purposes.

c) uses the same naming conventions for the test groups and test cases;

d) maintains the relationship, if any, specified in the TSS&TP between the test purposes and the entries in the PICS and partial PIXIT proformas to be used for test case selection;

e) complies with this Recommendation and with Recommendation X.292.

11 Specification of generic test suites

A conformance testing standard may include a generic test suite as a separate part, particularly if there is an intention to produce more than one ATS.

A generic test suite shall consist of one generic test case for each test purpose (or specified combination of test purposes), except for any which are not testable using the test methods defined in this Recommendation.

Note – Thus, a generic test suite has the coverage of the set of all possible ATSs for the relevant protocol(s). The test purposes excluded from a generic test suite will be testable only by using conformance resolution tests, which are not standardized.

Each generic test case adds detail to the test purpose. It should specify the main sequences of events of the test body and the verdicts to be assigned to the corresponding test outcomes.

Each generic test case specified shall be used as a common root of corresponding abstract test cases for different test methods.

If a generic test suite is produced in advance of ATSs, then it will be a useful step in the design process. If a generic test suite is produced after the production of at least one ATS, then it will provide a means of relating different test suites to one another and analysing where there may be gaps in their coverage.

Guidance on the production of generic test cases is provided in Annex D.

12 Abstract Test Methods

12.1 *Introduction*

An Abstract Test Method describes an abstract testing architecture consisting of a lower tester, upper tester and test coordination procedures, and their relationships to the test system and SUT. Each ATM determines the PCOs and test events (i.e. abstract service primitives (ASPs) and PDUs) which shall be used in an abstract test case for that ATM.

Each ATS shall be specified in accordance with one of the ATMs defined in this section.

12.2 *General principles*

12.2.1 *Lower Testers*

In all the ATMs, the lower tester communicates with the SUT via the appropriate underlying service provider. The physical medium is considered to be the service provider underneath the Physical layer.

The general specification of the ATMs, given in this section, refers to an IUT in which the highest layer is numbered “N_t” (for “top”) and the lowest is numbered “N_b” (for “bottom”). For single-protocol IUTs, N_t is equal to N_b. The same notation is used to refer to layers within the SUT and within the lower tester. The SUT may implement protocols in layers lower than "N_b", but these are not of interest in the test method descriptions. Nevertheless, the SUT shall include the Physical layer. For all test methods, ATSs specify test events at the lower tester point of control and observation (PCO) in terms of $(N_b – 1)$-ASPs and/or (N_b) to (N_t)-PDUs.

12.2.2 *Upper Testers*

The main difference between the ATMs is in the nature of the upper tester and its coordination with the lower tester.

In some test methods, a second PCO is employed, for the upper tester. In these test methods, the definition of the test events at the PCO for the upper tester shall be specified in accordance with the appropriate OSI service definition and OSI protocol specification. The activity at the upper tester PCO shall not require that the SUT or IUT support ASP parameters, PDUs or capabilities that are not part of an OSI CCITT Recommendation or International Standard.

If the PCO is at a humanly accessible interface, the SUT's user interface shall serve as the PCO.

12.3 *General specification of the ATMs for end-system IUTs*

12.3.1 *Introduction*

For IUTs within end-system SUTs there are four categories of ATMs: Local, Distributed, Coordinated and Remote.

12.3.2 *The Local test method*

Abbreviation: L

In this test method:

a) the test events at the lower tester PCO are specified only in terms of $(N_b – 1)$-ASPs and/or (N_b) to (N_t)-PDUs;

b) the test events at the upper tester PCO are specified in terms of (N_t)-ASPs;

c) the upper service boundary of the IUT shall be a standardized hardware interface which can be used for testing purposes; the test suites shall not place any requirements on the realization of the interface in the SUT, additional to those in the standardized hardware interface specification;

d) the specification of the hardware upper interface of the IUT shall define the mapping between the relevant ASPs and/or PDUs and their realization at the interface;

e) the upper tester is located within the test system;

f) the requirements for the test coordination procedures shall be specified in the ATS but are realized locally within the test system.

This test method is illustrated in Figure 1/X.291.

12.3.3 *The Distributed test method*

Abbreviation: D

In this test method:

a) the test events at the lower tester PCO are specified only in terms of $(N_b – 1)$-ASPs and/or (N_b) to (N_t)-PDUs

b) the test events at the upper tester PCO are specified in terms of (N_t)-ASPs;

c) the upper service boundary of the IUT shall be either a human user interface or a standardized programming language interface which can be used for testing purposes; the test suites shall not place any requirements on the realization of the interface in the SUT, additional to those in the standardized programming language interface specification, if applicable;

d) there shall be a mapping between the relevant ASPs and their realization at the upper interface of the IUT;

e) the upper tester is located within the SUT;

f) the requirements for the test coordination procedures shall be specified in the ATSs, although the procedures themselves shall not be;

g) if the upper interface of the IUT is a human user interface, then the human operator of the SUT fulfils the requirements of the TCPs;

h) if the upper interface is a standardized programming language interface, then the upper tester is realized in software and the upper and lower testers together fulfil the requirements of the test coordination procedures (TCP).

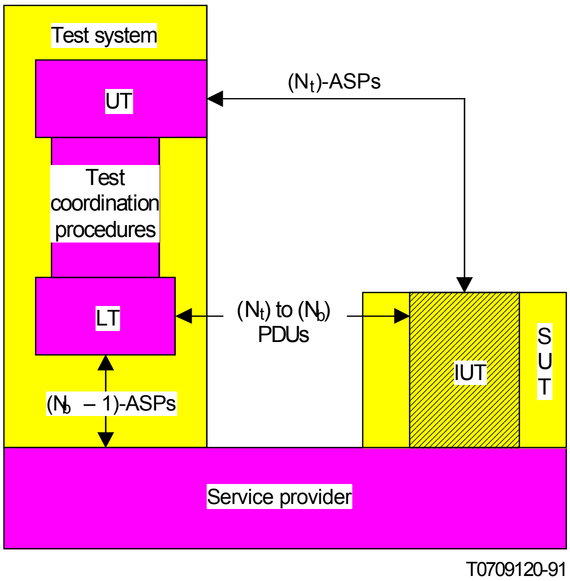


FIGURE 1/X.291

The local test method

This test method is illustrated in Figure 2/X.291.

ATSs for the Distributed test method shall not themselves specify an upper tester interface.

In order to avoid placing requirements on the internal design of SUTs, the ATSs shall not require that a programming language interface is standardized for the sole purpose of testing.

Note – In the Application layer, until application programming interfaces are standardized to provide a common means of access to OSI Application services, the use of the Distributed test method is in practice limited to using human user interfaces to OSI applications (e.g. File Transfer Access and Management initiators).

12.3.4 *The Coordinated test method*

Abbreviation: C

In this test method:

a) the test events at the lower tester PCO are specified in terms of $(N_b - 1)$-ASPs, and/or (N_b) to (N_t)-PDUs plus TM-PDUs;

b) (N_t)-ASPs are not used in the specification of the ATS; no assumption is made about the existence of an upper service boundary of the IUT;

c) the upper tester is located within the SUT;

d) the requirements for the test coordination procedures shall be specified in the ATS by means of a standardized test management protocol (TMP), referenced by the ATS;

e) the upper tester shall implement the TMP and achieve the appropriate effects on the IUT;

f) test cases shall be added to the ATS for the purpose of testing that the upper tester conforms to the requirements of the TMP specification; such test cases do not contribute to the conformance assessment of the IUT.

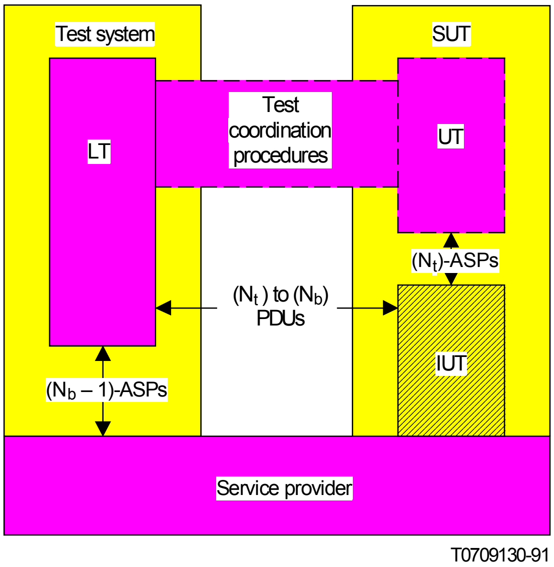


FIGURE 2/X.291

The distributed test method

A standardized TMP is applicable to a particular standardized ATS for the Coordinated test method and may not be applicable to other ATSs for the Coordinated test method.

Concerning the TMP:

a) the TMP shall be implemented within the SUT directly above the abstract service boundary at the top of the IUT;

b) the IUT shall not be required to interpret TM-PDUs, only pass them to and from the upper tester;

c) a TMP is defined only for testing a particular protocol and so does not need to be independent of the underlying protocol;

d) verdicts on test cases shall not be based on the ability of the SUT to exhibit any ASP or parameter of an ASP at the upper service boundary of the IUT, since this would contradict the definition of the Coordinated test method: the upper service boundary of the IUT is not a PCO in this test method. However, it is recommended that the TMP be defined separately from the ATS(s) in order to facilitate the task of the implementor of an upper tester. The definition of the TMP (as with the definition of any OSI protocol by ISO) can refer to the ASPs of its underlying service (i.e. the ASPs at the upper service boundary of the IUT).

This test method is illustrated in Figure 3/X.291.

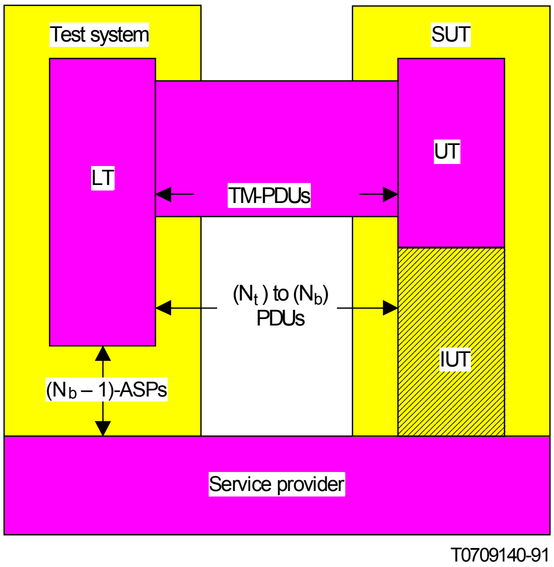


FIGURE 3/X.291

The coordinated test method

12.3.5 *The Remote test method*

Abbreviation: R

In this test method, provision is made for the case where it is not possible to observe and control the upper service boundary of the IUT. Also in this test method:

a) the test events at the lower tester PCO are specified only in terms of $(N_b – 1)$-ASPs, and/or (N_b) to (N_t)-PDUs;

b) (N_t)-ASPs are not used in the specification of the ATS; no assumption is made about the existence of an upper service boundary of the IUT;

c) some requirements for test coordination procedures may be implied or informally expressed in the ATS but no assumption shall be made regarding their feasibility or realization;

d) abstractly the SUT needs to carry out some upper tester functions to achieve whatever effects of test coordination procedures and whatever control and/or observation of the IUT are implied or informally expressed in the ATS for a given protocol; these functions are not specified nor are any assumptions made regarding their feasibility or realization;

e) the lower tester should attempt to achieve the implied or informally expressed test coordination procedures in accordance with the relevant information in the PIXIT.

In addition, in order to overcome the lack of specification of behaviour above the IUT, where necessary, the required behaviour of the SUT shall be specified in terms of the $(N_b – 1)$-ASPs or (N_b) to (N_t)-PDUs which need to be observed by the lower tester. This form of implicit specification shall be taken to mean "do whatever is necessary within the SUT in order to provoke the required behaviour".

However, it is possible that some of the test cases in the ATS cannot be executed (e.g. transmission of consecutive unacknowledged Data PDUs, etc.).

Even with such implicit specification of control of the IUT, in this test method it is possible to specify control but not observation above the IUT. This is a major difference between this and the other test methods.

This test method is illustrated in Figure 4/X.291.

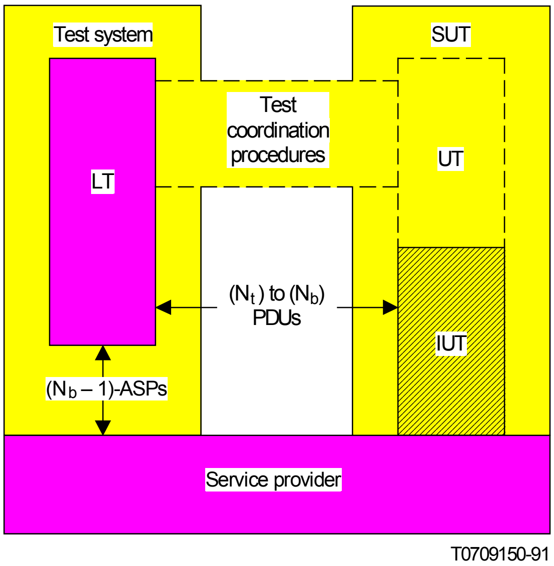


FIGURE 4/X.291

The remote test method

12.3.6 *Single-layer and embedded variants*

Each of the test methods has a variant which can be applied to single-protocol IUTs (abbreviation: S).

For a multi-protocol IUT, an embedded variant of each of the the test methods has been defined (abbreviation: E) to test the protocols one at a time.

If control and observation are applied to a means of access to the upper service boundary of the entities under test within the SUT, then the test methods are normal (and no E is added to the abbreviated name). If, however, control and observation are applied through one or more OSI protocol entities above the entities under test, the test methods are called embedded (and an E is appended to the abbreviated name).

Names of particular variants of the test methods shall be formed as follows:

{ L ; D ; C ; R } [;S ;SE;]

For example, DS is the abbreviation for the "Distributed Single-layer" test method, as defined in § 12.4.3, and DSE is the abbreviation for the "Distributed Single-layer Embedded" test method, as defined in § 12.5.3.

12.4 *Variants for single-protocol IUTs*

12.4.1 *Introduction*

In the following descriptions of the single-layer test methods, for testing single-protocol IUTs, the abstract model of the IUT is called the (N)-entity under test.

12.4.2 *The LS test method*

In the Local Single-layer (LS) test method, the test events are specified in terms of (N)-ASPs at the upper hardware interface of the (N)-entity under test, and (N – 1)-ASPs and/or (N)-PDUs at the lower tester PCO.

This variant is illustrated in Figure 5/X.291.

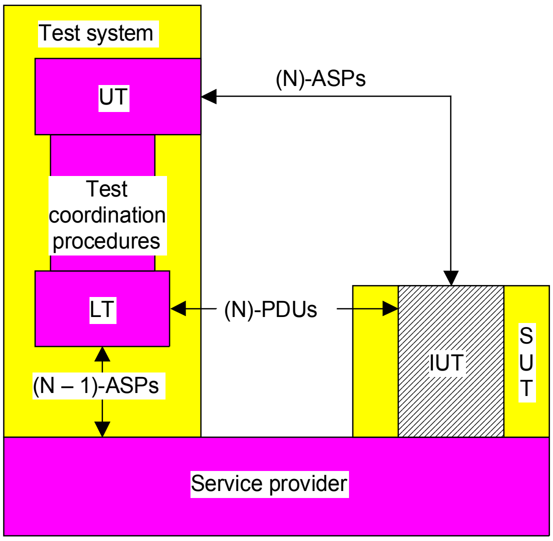


FIGURE 5/X.291

The LS test method

12.4.3 *The DS test method*

Also in the Distributed Single-layer (DS) test method, the test events are specified in terms of (N)-ASPs at the upper interface of the (N)-entity under test, and (N – 1)-ASPs and/or (N)-PDUs at the lower tester PCO.

Note – The DS test method differs from the LS test method in that the upper interface of the (N)-entity under test is not a hardware interface. This variant is illustrated in Figure 6/X.291.

12.4.4 *The CS test method*

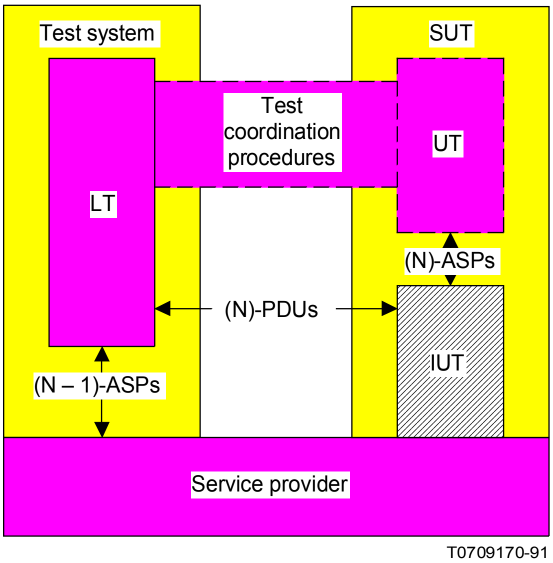
In the Coordinated Single-layer (CS) test method, the test events are specified in terms of (N – 1)-ASPs and/or (N)-PDUs, plus TM-PDUs, at the lower tester PCO.

This variant is illustrated in Figure 7/X.291.

12.4.5 *The RS test method*

In the Remote Single-layer (RS) test method, the test events are specified in terms of (N – 1)-ASPs and/or (N)-PDUs at the lower tester PCO.

This variant is illustrated in Figure 8/X.291.



Note – The DS test method differs from the LS test method in that the upper interface of the (N)-entity under test is not a hardware interface.

FIGURE 6/X.291

The DS test method

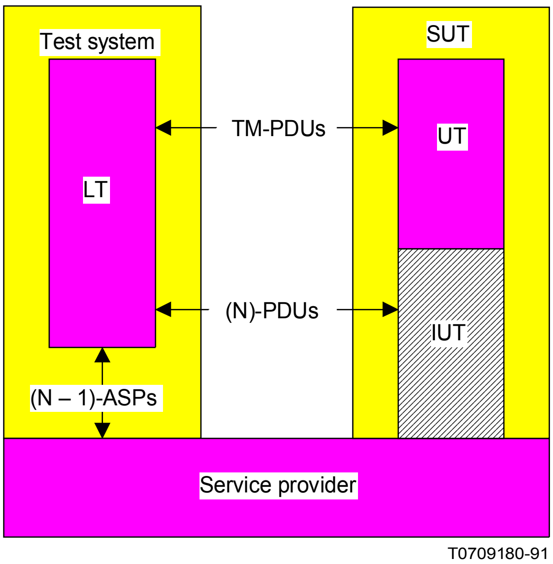


FIGURE 7/X.291

The CS test method

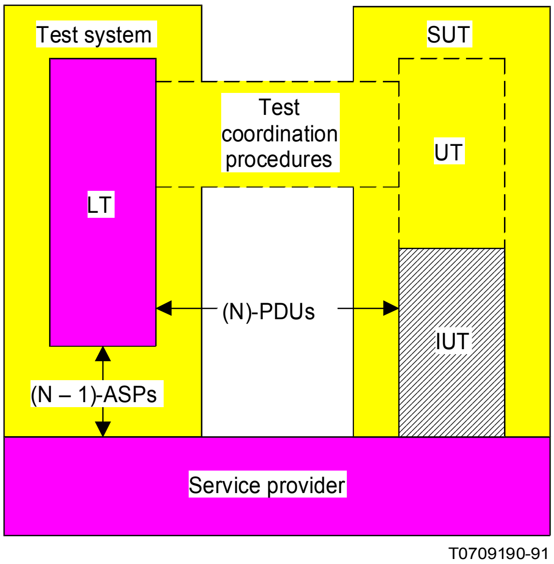


FIGURE 8/X.291

The RS test method

12.5 *Variants for multi-protocol IUTs*

12.5.1 *Introduction*

In single-layer embedded test method variants, testing is specified for a single-protocol within a multi-protocol IUT, including the specification of the protocol activity above the one being tested but without specifying control or observation at service boundaries within the multi-protocol IUT. Thus in a multi-protocol IUT from protocol (N_b) to (N_t), abstract test cases for testing protocol (N_i) shall include the specification of the PDUs in protocols ($N_i + 1$) to (N_t) as well as those in protocol (N_i)

Note – This description of the embedded test method variants assumes that the protocols of the IUT are ordered in a continuous adjacent user/ provider relationship.

Successive use of a single-layer embedded test method variant (from layer (Nb) to (Nt)) is called incremental testing of a multi-protocol IUT.

The embedded test method variants are defined for a single protocol under test in a multi-protocol IUT. This does not mean that there cannot be accessible service boundaries within the multi-protocol IUT: it means that no such boundaries are used in the test methods. Thus, all protocols between the protocol under test and the highest layer protocol for which PDUs are expressed as test events in the ATS shall be regarded as being part of the multi-protocol IUT.

Note – For the top layer in the multi-protocol IUT, (Nt), these variants are the same as the normal single-layer test methods.

12.5.2 *The LSE test method*

In the Local Single-layer Embedded (LSE) test method, for protocol (N_i) within a multi-protocol IUT from (N_b) to (N_t), the test events shall be specified in terms of the (N_t)-ASPs above the IUT and the (N_i – 1)-ASPs and (N_i) to (N_t)-PDUs above the (N_i – 1)-service provider in the test system.

This variant is illustrated in Figure 9/X.291.

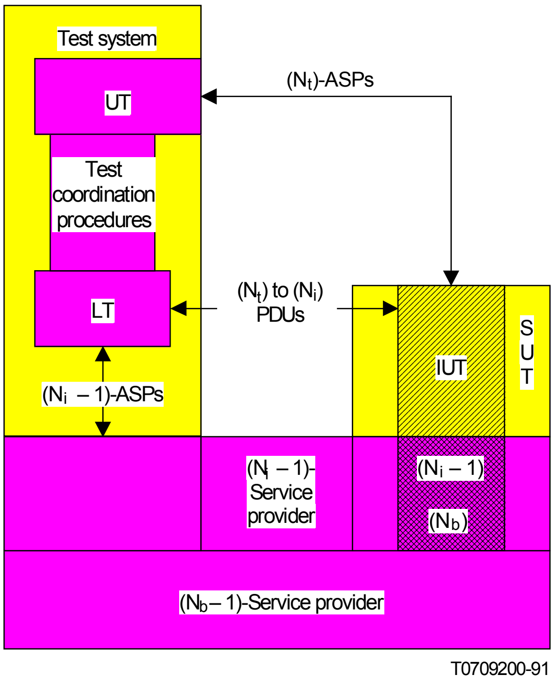


FIGURE 9/X.291

Example of the LSE test method: testing the (N_i)-protocol in an (N_b) to (N_t)-protocol IUT

12.5.3 *The DSE test method*

In the Distributed Single-layer Embedded (DSE) test method, for protocol (Ni) within a multi-protocol IUT from (N_b) to (N_t), the test events shall be specified in terms of the (N_t)-ASPs above the IUT and $(N_i - 1)$-ASPs and (N_i) to (N_t)-PDUs above the $(N_i - 1)$-service provider in the test system.

This variant is illustrated in Figure 10/X.291.

12.5.4 *The CSE test method*

The Coordinated Single-layer Embedded (CSE) test method uses features of both the CS and the DSE test methods. The test events shall be specified in terms of $(N_i - 1)$-ASPs, (N_i) to (N_t)-PDUs, and TM-PDUs. The TMP shall be designed to operate over the (N_t)-Service.

This variant is illustrated in Figure 11/X.291.

12.5.5 *The RSE test method*

The Remote Single-layer Embedded (RSE) test method uses the same PCO as the RS test method for the same layer, but differs from the RS test method in that $(N_i + 1)$ to (N_t)-PDUs shall be specified in test cases for layer (N_i).

This variant is illustrated in Figure 12/X.291.

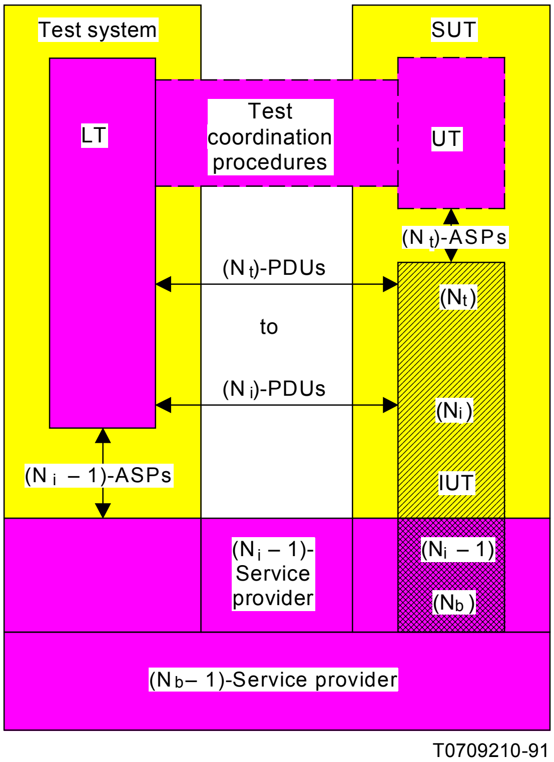


FIGURE 10/X.291

Example of the DSE test method:
testing the (N_i)-protocol in an
(N_b) to (N_t)-protocol IUT

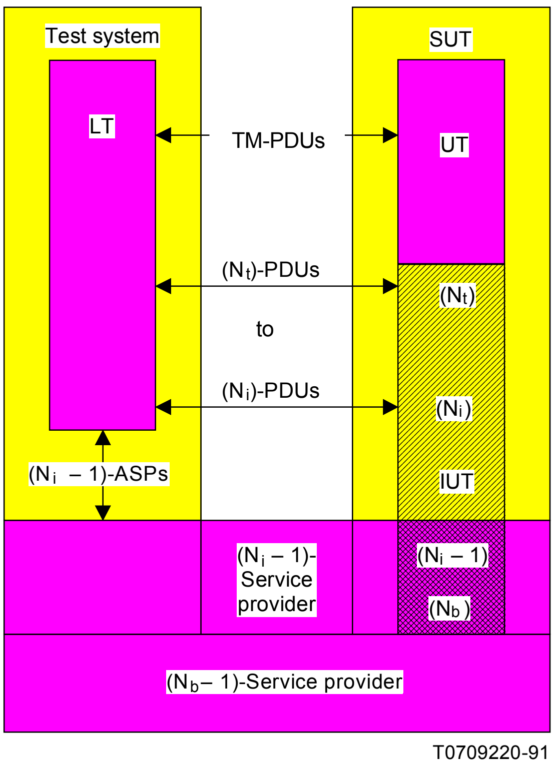


FIGURE 11/X.291

Example of the CSE test method:
testing the (N_i)-protocol in an
(N_b) to (N_t)-protocol IUT

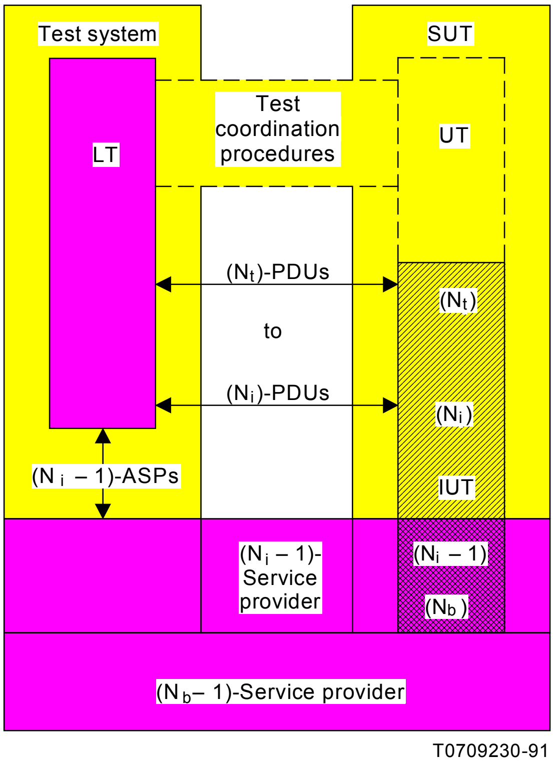


FIGURE 12/X.291

Example of the RSE test method: testing the (N_i)-protocol in an (N_b) to (N_t)-protocol IUT

12.6 *ATMs for open relay-systems*

12.6.1 *Introduction*

For open relay-systems, loop-back and transverse ATMs are defined. They are given abbreviated names: YL and YT respectively.

12.6.2 *The YL test method*

The YL test method is used for testing a relay system from one subnetwork

This test method is illustrated in Figure 13/X.291.

For this test method there are two PCOs on one subnetwork at SAPs external from the (N_t)-Relay. For connection-oriented protocols, it requires that the two test connections are looped together on the far side of the (N_t)-Relay, but it is not specified whether this looping is performed within the (N_t)-Relay or in the second subnetwork. For connectionless protocols, it requires that the PDUs are looped back within the second subnetwork and addressed to return to the second PCO.

This method enables an open relay-system to be tested without requiring test systems on two different subnetworks. Thus, the procedures for coordinating the control applied to the 2 PCOs can be realized within a single test system.

The disadvantage of this test method is that the behaviour of the relay on only one side is directly observed. Thus, its behaviour on the second subnetwork cannot be properly assessed.

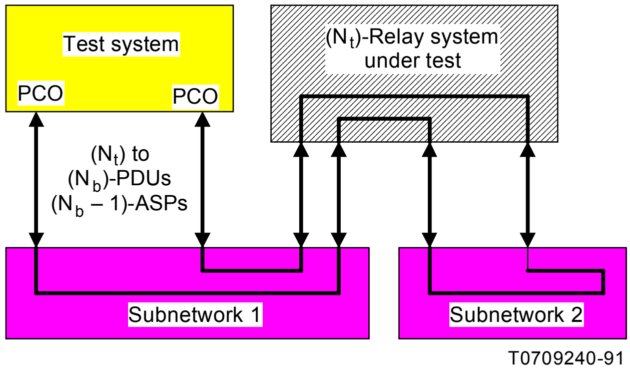


FIGURE 13/X.291

Loop-back test method (YL)

12.6.3 *The YT test method*

The YT test method is used for testing a relay system from two subnetworks.

This test method is illustrated in Figure 14/X.291.

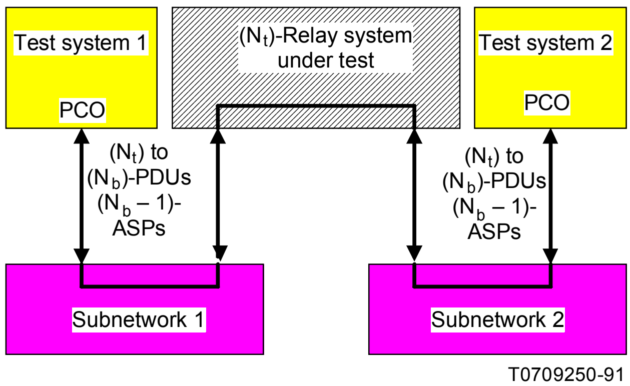


FIGURE 14/X.291

Transverse test method (YT)

In this test method there are 2 PCOs, one on each subnetwork, at SAPs external from the (N_t)-Relay.

This method enables an open relay-system to be tested in its normal mode of operation, with its behaviour on each subnetwork being observed.

12.7 *Choice of ATM*

12.7.1 *Introduction*

Before an ATS can be defined, it is necessary to study all the environments in which the protocol is likely to be tested and establish accordingly the ATM(s) to be used for the production of one or more ATSs.

ATMs vary in the extent of control and observation of an IUT that they provide. The choice of test method, therefore, influences the expressibility of the behaviour in test case descriptions.

12.7.2 *Comprehensive testing service*

ATS specifiers shall place a requirement in the conformance testing standard defining which ATM(s) shall be supported as a minimum by any organization claiming to provide a comprehensive testing service for the protocol(s) in question. If an organization supports this minimum set of ATMs, then it may claim to provide a comprehensive testing service even when other ATMs may be applicable to the protocol(s) in question.

A comprehensive testing service shall include at least one test method which places no additional requirements upon the SUT other than those contained in the OSI CCITT Recommendations or International Standards to which the SUT claims to conform.

To meet this requirement a test suite for the Remote test method shall be included in a comprehensive testing service, unless one of the other test methods also meets this requirement. For some protocols in, or embedded under, the Application layer, it may be possible to meet this requirement by including a test suite for the DSE test method. For IUTs with hardware upper interfaces it may also be possible to meet the requirement by including a test suite for the Local test method.

If a standardized ATS is produced which does not meet the above requirement for provision of a comprehensive testing service, then it shall contain the following statement in the Scope section:

"This abstract test suite is insufficient on its own for the provision of a comprehensive testing service for the <*name*> protocol."

A comprehensive testing service requirement statement shall be located, as a separate section in the part of the conformance testing standard containing the test purposes for a particular protocol.

12.7.3 *IUT environments*

There is a relationship between the test methods and the configurations of the real open systems to be tested.

Recommendation X.290 (1992), § 7.2, gives a full account of the classification of systems and IUTs.

When choosing a test method, the test suite specifiers shall identify, if they have not already done so, whether the test suites they plan to produce are for IUTs which are single-layer and:

a) belong to end or relay systems;

b) belong to complete or partial systems;

c) belong to fully open or mixed systems;

d) have service boundaries accessible or not;

e) are special purpose (i.e. to be used by a single application) or general purpose (i.e. to be used by several applications).

12.7.4 *Applicability of the ATMs*

Some considerations concerning the applicability of the test methods to different layers are discussed in Recommendation X.290 (1992), Annex B.

One or more appropriate ATMs shall be selected for the protocol being considered.

For each protocol for which ATSs are to be produced, priorities should be assigned to the standardization of different ATSs for different applicable test methods, giving highest priority to those that are most likely to be applicable to the majority of real systems.

12.8 *Test coordination procedures*

For effective and reliable execution of conformance tests, some set of rules is required for the coordination of the test process between the lower tester and the upper tester. The general objective of these rules is to enable the lower tester to control the operation of the upper tester, in ways necessary to run the test suite selected for the IUT.

These rules lead to the development of test coordination procedures to achieve the synchronization between the lower tester and the upper tester and the management of information exchanged during the testing process. The details of this synchronization and how the required effects are achieved, are closely related to the characteristics of the SUT as well as to the test methods.

The requirements on the test coordination procedures shall be specified for each ATS. The test coordination procedures shall include provision for relaying, to the lower tester, events which are controlled, (and if applicable, observed) at the upper tester, and which need to be logged.

When defining test cases for the Local and Distributed test methods, requirements on the upper tester and/or test coordination procedures which may be necessary shall not exceed those identified in §§ 12.3.2 and 12.3.3 for the Local and Distributed test methods, respectively.

13 Specification of ATSs

13.1 *General*

An ATS comprises a set of test cases and optionally test steps for a particular test method.

Preceding the test cases themselves shall be the following information:

a) ATS name, date of origin and version number;

b) names (and version numbers) of the CCITT Recommendation(s) or International Standard(s) which specify the protocol(s) (and if relevant, the transfer syntax) which are used in the specification of the test cases;

c) names (and version numbers) of the CCITT Recommendation(s) or International Standard(s) which define the OSI service(s), the ASPs of which are specified in the test cases as being controlled and/or observed;

d) name (and version number) of the CCITT Recommendation or International Standard defining the test notation;

e) name of target test method;

f) description of the coverage of the test suite; for example, the functional subsets of the protocol(s) that are tested;

g) description of the structure of the test suite, in terms of test groups and their relationship to the protocol specification(s);

h) description of the test coordination procedures or a reference to the specification of the TMP (if applicable in the test method);

i) optionally, a list of which capability and behaviour test cases may be used as basic interconnection tests;

j) information to assist the test realizer and test laboratory in their use of the standardized ATS (see § 15);

k) an identification of the Technical Corrigenda (or CCITT equivalent) which are related to the CCITT Recommendation or International Standard that specifies the protocol or transfer syntax, and which have been taken into account in the ATS.

13.2 *Test cases*

13.2.1 The ATS specifier shall apply a standardized test notation in which to specify the abstract test cases. TTCN, defined in Recommendation X.292, is recommended for this purpose.

13.2.2 If a standardized ATS uses facilities additional to those in TTCN as defined in Recommendation X.292, then such additions shall be documented in the standardized ATS and submitted for inclusion in Recommendation X.292 by means of defect reports or an amendment, as appropriate.

13.2.3 Once the test notation and test method have been chosen, the abstract test cases can be specified.

Each abstract test case shall:

a) reflect only a single test purpose, or a combination of test purposes as defined by the test purpose specifier, or if there is one available, reflect a single generic test case;

b) specify all the sequences of test events that comprise a test body;

c) specify all sequences of test events that comprise the test preamble(s), if any, necessary to ensure that it is capable of being started in the idle testing state and, optionally, in one or more other stable testing states (see § 13.2.4);

d) specify all sequences of test events that comprise the test postamble(s), if any, necessary to ensure that it is capable of ending in the idle testing state and, optionally, in one or more stable testing states;

e) be specified using the chosen test notation and test method;

f) specify the test verdict to be associated with each possible sequence of test events comprising a complete path through the test case.

13.2.4 If a test purpose can be achieved only by means of system-dependent actions in the SUT, it is not possible to specify an abstract test case for that test purpose in a standardized ATS. This limitation shall be documented in the standardized ATS.

Note – The possibility of writing ad hoc conformance resolution tests to achieve the test purpose on a case-by-case basis should be indicated, but such tests are outside the scope of standardization.

If a test purpose cannot be achieved due to the specific nature of the chosen ATM, that limitation shall also be documented in the standardized ATS.

Thus, for each specified test purpose the standardized ATS shall either specify an abstract test case to achieve that test purpose, or shall document why such a test case is not included.

13.2.5 A choice of more than one test preamble may be specified in a given abstract test case, one for each of the stable testing states in which the test case can start. Each test preamble takes the test case from a particular stable testing state to the initial testing state of the test body. Thus, a small set of stable testing states, in which test cases may start and end, shall be defined for the ATS; this set shall include the appropriate idle testing state.

Note – It is likely that not more than two or three test preambles will need to be used.

In each abstract test case in which the initial testing state of the test body is not the idle testing state, the test suite specifier shall define a test preamble to take the test case from the idle testing state to the initial testing state of the test body. In addition, in each abstract test case in which the test body does not necessarily end in the idle testing state, the test suite specifier shall define one or more test postambles to enable the abstract test case to end in the idle testing state.

Note – The ability to start and end an abstract test case in an idle testing state is necessary in order to be able to run each abstract test case individually, in isolation from the other abstract test cases.

If more than one test preamble or postamble is defined for an abstract test case, then the test suite specifier shall specify the conditions under which each test preamble or postamble is to be used. The choice of test preamble shall depend upon the stable testing state in which the test case starts. The choice of test postamble shall depend upon the testing state in which the test body ends and the stable testing state in which the test case is to end.

The omission of a test preamble from an abstract test case shall be permitted only if the initial testing state of the test body is the desired starting stable testing state of the test case. Similarly, the omission of a test postamble from an abstract test case shall be permitted only if the final testing state of the test body is one of the desired ending stable testing states of the test case. Each test postamble takes the test case from the end of the test body to a stable testing state in which the test case can end.

If it is the intention to be able to make use of test preambles that start in some stable testing state other than the idle testing state, the test suite specifier shall specify that the identity of the ending stable testing state of each abstract test case is stored for access by the next test case. The next test case can then compare the identity of that state with the possible stable testing states, in order to determine which test preamble to use. In this way, the use of test preambles is made conditional on the starting stable testing state, not unconditionally optional.

If the initial testing state of the test body is a transient testing state, then the test body shall not be run without first running a test preamble.

13.2.6 Each test preamble, test body and test postamble may be explicitly identified as test steps, but they need not be.

In designing the test step structure within abstract test cases, the test suite specifier can benefit from using the same test steps in several abstract test cases.

13.2.7 The test suite specifier shall ensure that each abstract test case explicitly defines:

a) each sequence of test events to be associated with a "pass" verdict;

b) each sequence of test events to be associated with an "inconclusive" verdict;

Note – This verdict would be associated with sequences of test events representing behaviour from the IUT which although valid prevents the test purpose(s) from being accomplished.

c) all remaining sequences of test events to be associated with a "fail" verdict, either defined individually or categorized by using an unidentified test event.

13.2.8 The checking that is to be performed in a test case, for the validity of test events received from the IUT with respect to the relevant protocol specification(s), shall be specified explicitly within the abstract test case. The test suite specifier shall not assume that the test realizer or test laboratory will perform any checking of the test events against the protocol specification(s) other than that which is specified in the abstract test cases.

13.3 *ATS conformance clause*

The standardized ATS shall include a conformance clause.

The conformance clause shall contain the following statement:

"The test realizer shall comply with the requirements of Recommendation X.293. In particular, these concern the realization of an ETS based on the ATS.

Test laboratories running conformance test services for this abstract test suite shall comply with Recommendation X.294."

13.4 *Consistency with protocol*

A standardized ATS shall represent accurately the protocol(s) that it tests conformance to. If errors or ambiguities are discovered in the protocol specification during development of the ATS, the test suite specifier shall forward, to the proper CCITT or ISO/IEC group, defect reports which identify the problems. If differences are discovered between an ATS and the protocol specification after the ATS is standardized, then the protocol specification shall have precedence in problem resolution.

Note – FDTs may facilitate validation of a test suite against a protocol specification.

14 Specification of a TMP

In the case of the Coordinated test method (CS and CSE) the test coordination procedures are realized by the standardization of a TMP, as a separate part of the conformance testing standard.

The TMP needs to be able to convey requests to the IUT to achieve the effect of a service primitive and to convey back to the lower tester the record of observations of effects equivalent to the occurrence of service primitives. The upper tester should be an implementation of the TMP. Test cases shall be added to the ATS for the purpose of testing that the upper tester conforms to the requirements of the TMP specification. Such test cases do not, however, contribute to the conformance assessment of the IUT.

If a TMP part of the conformance testing standard is developed, then a proforma shall be provided for a TMP implementation statement which shall include an entry for each of the TM-PDUs.

15 Use of an ATS specification

15.1 The ATS specifier shall provide information in the standardized ATS to assist the test realizer and test laboratory in their use of the test suite. This information shall include, but is not limited to, the following:

a) a mapping of the abstract test cases to the PICS proforma entries to determine whether or not each abstract test case is to be selected for the particular IUT; the mapping should be specified in a notation suitable for Boolean expressions;

b) the specification of a partial PIXIT proforma for each ATS; this proforma shall contain a list of all parameters for which the test suite requires values; if any of the required parameter values are to be found in the PICS, the PIXIT proforma entry for each such parameter shall reference the corresponding entry in the PICS proforma;

Note – Other aspects of the PIXIT proforma are discussed in Recommendations X.290, X.293 and X.294.

c) a mapping of the abstract test cases to the partial PIXIT proforma, to parameterize the test suite for the particular IUT; the mapping shall identify requirements for testing which may prevent test cases from being run against a particular IUT; the mapping should be specified in a notation suitable for Boolean expressions;

d) the order in which the abstract test cases are to be listed in the PCTR (see § 15.2);

e) any restrictions that there may be on the order in which the test cases may be executed;

f) identification of test cases or test groups which shall be realized in an MOT claiming to conform to the standardized ATS;

g) the requirements on the test coordination procedures or a reference to the specification of the TMP (if applicable in the chosen test method);

h) any necessary timing information.

15.2 The order in which the abstract test cases are to be listed in the PCTR may be specified explicitly in the standardized ATS as a list, or implicitly (by default) as the order in which the abstract test cases are specified in the standardized ATS. In addition, the standardized ATS may provide information on the status of each test case which shall be preserved in the PCTR.

If any listed basic interconnection tests are run as a preliminary stage in the conformance assessment process, the test verdicts associated with them shall be listed in the PCTR in the positions indicated for the corresponding capability or behaviour test cases (i.e. as if they were run as capability or behaviour tests).

15.3 The order in which the abstract test cases are listed in the ATS does not imply a precise order of execution. However, restrictions may be specified on the possible orders of execution (i.e. defining a partial ordering, e.g. it may be desirable to run a simple abstract test case before running more complex and detailed variants of that test case).

Note – Optimization of the order of execution of test cases in order to minimize execution time is considered to be a performance matter. This area is outside the scope of standardization.

16 ATS maintenance

Once an ATS has been specified and is in use, it can be expected that errors and omissions in it will be detected by those who are using the test suite. The ATS specifier shall in such circumstances progress the updating of the test suite via the relevant defect reporting procedures.

In addition, from time to time, changes will be made to the protocol specification(s) to which the test suite relates. The ATS specifier shall ensure that the test suite is updated as soon as possible after changes to the relevant protocol specification have been ratified.

ANNEX A

(This annex forms an integral part of this Recommendation)

Requirements and guidance for a PICS proforma

A.1 *Introduction*

A.1.1 A PICS proforma defines explicitly the implementation flexibility allowed by the protocol specification. It details in a tabular form:

a) the implementation options, i.e. the functions additional to those which are mandatory to implement; and

b) the legitimate range of variation of the global parameters controlling the implementation of the functions, as specified in the protocol specification.

A.1.2 For a specific protocol, the PICS proforma is used by:

a) the implementors or suppliers, who need to document their implementations;

b) ATS specifiers, who need to ensure that the structure of the test suite matches the allowed implementation flexibility;

c) the specifiers of OSI protocol profiles, who require a detailed definition of the implementation flexibility available in each base protocol specification.

A.1.3 A completed PICS proforma is the PICS for the implementation in question. A PICS makes a contribution to the conformance assessment process, where it is used in:

a) the static conformance review;

b) the test selection process, as a means of adapting Executable Test Suites to the options supported by the implementation;

c) the results analysis process, as a reference document.

PICSs can also be used to assess the ability of two implementations to interwork. This can be done by a comparison of the options and parameters stated in the PICSs.

A.1.4 Each protocol-defining group is responsible for the technical content of the PICS proforma(s) related to the protocol(s) it maintains.

This annex provides requirements and guidance on how a PICS proforma should be structured and on what questions it should contain. It is not possible to provide a generic PICS proforma, because of the wide variety of protocols. Nevertheless, some general rules are applicable to any OSI protocol specification.

A.2 *Relationship between PICS proformas and conformance requirements*

A.2.1 A PICS proforma is a set of questions related to the capabilities of the protocol. A protocol capability is a set of functions which is to be supported by an implementation. The static conformance requirements of the OSI protocol specification define rules for the implementation of the capabilities.

Each PICS proforma question (or entry) shall indicate a status for each capability, according to those rules.

This status can be:

a) mandatory – the capability is required to be implemented, in conformance with the protocol specification;

b) optional – the capability may be implemented, and if it is implemented it is required to conform to the protocol specification; options can be Boolean, mutually exclusive, or selectable (as described in Recommendation X.290, § A.3);

c) prohibited – there is a requirement not to use this capability in a given context (applicable only to dynamic requirements, if any, embedded in a PICS proforma);

d) not applicable – no requirement can be expressed in a given context;

e) conditional – the requirement on the capability depends on the selection of other optional or conditional items; the PICS proforma cannot define in advance a definite status for the capability, it can only indicate that the status (mandatory, optional, prohibited, or non-applicable) depends on the evaluation of a predicate or on a conditional expression.

A.2.2 A PICS proforma entry shall provide space to record the statement of the supplier of the IUT regarding support of the capability in the implementation. The support may be recorded as:

a) the capability is implemented;

b) the capability is not implemented;

c) other protocol-specific categories of support.

Note – It is possible that the static conformance requirements do not provide a fully detailed definition of the implementation flexibility rules for the protocol. Additional rules may be found embedded in the body of the protocol specification.

A.2.3 When a mandatory capability is not supported, it is a case of non-conformance. See § A.7.

When an optional capability is not supported, a question may be asked to ascertain what action is taken by the implementation when PDUs related to the capability are received, depending on whether the protocol specification

a) specifies a choice of actions that may be taken, or

b) does not specify what action may be taken.

A.2.4 There shall be a well-defined mapping (by references) from the PICS proforma to the static conformance requirements, as stated in § 7.3.

A.2.5 The PICS proforma status entries implicitly define the checking to be performed during the static conformance review. The PICS proforma may also define additional specific checks to be performed during the static conformance review (see § A.8.10).

A.3 *General layout*

A.3.1 The PICS proforma shall be produced as a distinct CCITT Recommendation or as a normative part of the relevant OSI protocol International Standard. The relevant (CCITT or ISO/IEC) rules for structuring a CCITT Recommendation or International Standard apply.

A.3.2 The Introduction clause shall include the following text:

"To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given OSI protocol. Such a statement is called a Protocol Implementation Conformance Statement (PICS)."

A.3.3 The Scope clause shall contain the following text:

"This [CCITT Recommendation/International Standard] provides the PICS proforma (for the <name> protocol as specified in <reference>) in compliance with the relevant requirements, and in accordance with the relevant guidance, given in Recommendation X.291."

A.3.4 The References clause shall contain the following references:

"Rec. X.290 (1992) – *OSI Conformance Testing Methodology and Framework for Protocol Recommendations for CCITT Applications – General concepts*. (See also ISO/IEC 9646-1: 1991)"

"Rec. X.291 (1992) – *OSI Conformance Testing Methodology and Framework for Protocol Recommendations for CCITT Applications – Abstract Test Suite specification.* (See also ISO/IEC 9646-2: 1991)"

The References clause shall also contain a reference to the relevant CCITT Recommendation or International Standard which specifies the protocol.

A.3.5 The Definitions clause shall contain:

"This [CCITT Recommendation/International Standard] uses the following terms defined in Recommendation X.290:

a) PICS proforma;

b) protocol implementation conformance statement (PICS);

c) static conformance review."

Note – Item c) is needed only if the PICS proforma actually mentions the static conformance review (see § A.8.10).

A.3.6 A clause should be included to refer to the protocol conformance requirement concerning the PICS, stating:

"The supplier of a protocol implementation which is claimed to conform to <reference> is required to complete a copy of the PICS proforma provided in Annex <X> and is required to provide the information necessary to identify both the supplier and the implementation."

In addition, the Conformance clause of the protocol specification shall contain:

"The supplier of a protocol implementation which is claimed to conform to this <CCITT Recommendation or International Standard> shall complete a copy of the PICS proforma provided in <reference to PICS proforma part>, Annex <X>, and shall provide the information necessary to identify both the supplier and the implementation."

A.3.7 The body of the PICS proforma should be provided in an annex. The annex shall contain the actual proforma to be filled in by the supplier or client of a test laboratory. The following sections specify requirements and provide guidance for such a PICS proforma annex.

A.4 *Copyright*

PICS proformas are intended to be completed by the implementors in the form printed in the appropriate CCITT Recommendation or International Standard (copied or replicated). This raises an issue of copyright in respect of the text of that part of a CCITT Recommendation or International Standard.

The following statement shall appear in the PICS proforma annex, as a footnote on the first page, referenced from the title of the annex (e.g. "Annex A"):

"Copyright release for PICS proformas."

"Users of this Recommendation may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS."

The terms can be suitably modified to reflect the exact form of publication, e.g. "Recommendation" can be replaced by "International Standard" or "Technical Report".

Also the words "Unless otherwise specified," shall be added before "no part of this publication may be reproduced ..." in the copyright statement at the end of the Contents page(s).

A.5 *First section – Identification of the implementation*

The first section of the actual PICS shall identify the implementation and the supplier or client of a test laboratory.

For administrative purposes, the PICS itself shall include a cover page identifying:

a) the implementation and the system in which it resides;

b) the supplier of the system and/or the client of the test laboratory that is to test the implementation;

c) the person to contact if there are any queries concerning the content of the PICS;

d) the relationship of the PICS with the System Conformance Statement for the system.

It is not necessary for the PICS proforma to give a precise format for a table for such information. It shall, however, state the need for such information and should do so in the style of the above paragraph.

Note – A test laboratory may provide a proforma for the cover page.

A.6 *Second section – Identification of the protocol*

The second section identifies the CCITT Recommendation or the International Standard to which the PICS proforma may be applied. This includes the CCITT or ISO/IEC reference number, and the complete title. This section shall be included in the PICS proforma.

Different versions to which the PICS proforma may be applied should be identified explicitly, together with a status and support column if appropriate. If the OSI protocol provides a version parameter, then the second section shall reference another entry in the PICS Proforma, where detailed information on status and support of such a parameter (and possibly its negotiation) is given.

A.7 *Global statement of conformance*

A question shall be included in the PICS proforma to ask whether or not all mandatory capabilities are implemented.

A note shall be added to include the sense of the following:

"Answering "No" to this question indicates non-conformance to the protocol specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming."

A.8 *Other sections – Capabilities*

A.8.1 *Introduction*

The remaining sections shall list the protocol capabilities, data elements and parameters (insofar as questions related to those items are necessary to express the implementation flexibility allowed by the protocol).

The following topics are common to many OSI protocols, but they need to be adapted to each particular protocol in order to design the appropriate sections of the PICS proforma.

A.8.2 *Initiator/responder capability*

PICS proforma entries may be used to specify whether the implementation is able to initiate communication, respond to communication initiated by another system, or both. The way the PICS proforma entries are organized to handle the two modes (e.g. two sets of columns per table, or two sets of tables) shall be indicated to the reader. Such entries are recommended for asymmetric protocols.

A.8.3 *Major capabilities*

The full functionality of the protocol may be divided into large implementation "blocks" (i.e. functional units, service classes, service elements, protocol classes). If so, for each "block" a relevant PICS proforma entry shall give the conformance status of the capability, and provide space for the implementor support statement. Such entries shall be included in a separate section in the PICS proforma (unless the protocol contains only one such "block").

These “blocks” are referred to as major capabilities. They are defined (with their status) in the static conformance requirements of the relevant protocol specification. Depending on the nature of the protocol, the major capabilities may correspond to a group of PDUs to be implemented together, or to some global aspect common to several PDUs.

The dynamic conformance requirements corresponding to each major capability are not reproduced in the PICS proforma.

A major capability is supposed to be implemented in its entirety. Exceptions to this rule shall have a technical justification.

The PICS proforma shall have an entry for each major capability, whatever its status.

A.8.4 *Timers and protocol parameters*

PICS proforma entries may be used to list all timers and protocol parameters defined in the protocol specification. Allowed or mandatory lengths, data types and values (or range of values) shall be specified for each. Space should be provided for specifying which element or values are supported. Such entries are recommended whenever relevant.

A.8.5 *PDUs*

Entries to identify PDU support shall be included in the PICS proforma. These shall cover all PDUs defined for the protocol, grouped according to the major capabilities whenever relevant. In addition, the status and support for sending and receipt of each PDU should be indicated separately (see § A.8.2).

Note – The conformance clause may or may not provide information on the optional status of specific protocol elements (PDUs, PDU parameters). In some protocols, the optional status of some protocol elements is located in the body of the specification (dynamic conformance requirements), in others they are included in the conformance clause.

A.8.6 *PDU parameters*

PICS proforma entries may be used to list, for each PDU type, the parameters for which implementation flexibility exists. Such entries are recommended whenever relevant.

Note – If a parameter has always to be present, has always to be of a specific type, and has always to support a defined range of values, it need not be documented in the way indicated below.

For each documented parameter, the PICS proforma should provide:

a) its status, based on the value of a specified predicate in each direction (i.e. sending and receipt);

b) space to indicate whether or not it is supported in each direction;

c) the lengths, ranges of values and/or data types permitted in each direction by the relevant protocol or transfer syntax specification;

d) space to indicate the values supported in each direction.

Regarding the range of values, two sorts of PDU-parameters exist: those with and those without implementation flexibility.

If there is no implementation flexibility, only one question need be asked by the PICS proforma: is the parameter supported, with its full range of values?

If there is implementation flexibility, additional questions shall be asked in the PICS proforma. As an example, an “unlimited” PDU parameter calls for a question in the PICS proforma asking what is the maximum size implemented.

The proforma should give a clear indication of the preferred data types to be used for specifying the supported values (e.g. number bases, string types, octets, bits, seconds, etc.).

Other categories of PICS entry may also be used to cover the implementation flexibility regarding encoding rules.

For a protocol using a transfer syntax that does not strictly define the size of the parameters transferred (e.g. ASN.1), it should be made clear whether or not the sizes defined include the encoding.

A.8.7 *Negotiation capabilities*

PICS proforma entries may be used to describe the negotiation options available in the protocol, and provide space to indicate which have been implemented. Such entries are recommended whenever relevant.

A.8.8 *Protocol error handling*

If the protocol specification allows more than one method of error handling, PICS proforma entries may be used to list what these methods are, and provide space to indicate which are supported. Such entries are recommended whenever relevant.

A.8.9 *Multi-layer dependencies*

Whenever relevant, there should be PICS proforma entries to identify the implementation support for specific requirements on the underlying layers, not made mandatory by the underlying layer protocol specifications.

A.8.10 *Other conditions*

If there is a complex relationship between options which need to be checked in the static conformance review, and which cannot be associated with a specific PICS proforma entry, then such relationships should be documented in the form of Boolean expressions, or matrix tables in a separate section, with predicates and variables referencing the relevant PICS proforma entries.

A.9 *Formats of the tables*

A.9.1 *Structure of the tables*

The individual sections of the PICS proforma shall be presented in the form of one or more tables. The structure of these tables should follow the structure of the static conformance requirements and of the topics given in § A.8 above.

The tables shall list all the capabilities and options. There should be one item of the list per row.

Each row shall cross the following columns:

a) a pre-printed column to the left to give a reference number to each row;

   the reference number column shall provide a means of uniquely referencing each possible response within the PICS proforma. The means of referencing individual responses should be to specify the following sequence:

   1) a reference to the smallest subclause enclosing the relevant item,

   2) a solidus character, "/",

   3) the reference number of the row in which the response appears,

   4) if, and only if, more than one response occurs in the row identified by the reference number, then each possible entry is implicitly labelled a, b, c, etc., from left to right, and this letter is appended to the sequence;

b) one pre-printed column to name the item for each row;

c) one or more sets of columns to specify the status and record the support of the item; one set per distinct context in which the support is to be specified (e.g. for sending and receiving); each such set of columns may contain:

   1) a "status" column (pre-printed) to specify the status of the item (e.g. mandatory, optional, conditional), as defined in the relevant protocol or transfer syntax specification (status information is mandatory, but the column itself may be omitted if all items in a given table have the same status);

   2) another column, if appropriate, to specify the predicate upon which a conditional status is based (see also A.9.2.5 and A.9.2.6);

3) a pre-printed column giving references to the appropriate static conformance requirements or other clauses in the relevant protocol or transfer syntax specification(s) (mandatory to provide suitable references, preferably in such columns);

4) a "support" column, in which a response can be made to indicate whether or not the implementation supports the item in the particular context (mandatory column); if appropriate, space may be provided to indicate, in each context, the kind of "non-support" which is implemented, such as "receive and ignore", "receive and reject", etc.;

5) an "allowed values" pre-printed column, if appropriate, stating any restrictions or prescriptions on the types/lengths/ranges of values to be supported, as specified in the relevant protocol or transfer syntax specification;

6) a "supported values" column, if appropriate, in which the values or ranges of values supported can be indicated, as well as types and lengths, if relevant;

7) space on the right in which additional columns can be added if necessary to enable comments to be added by the user of the PICS proforma.

Figure A-1/X.291 gives examples of possible realizations of the tables.

A.9.2 *Symbols and conventions*

A.9.2.1 The standardized symbols for the status column are as follows:

a) m or M for mandatory;

b) o or O for optional (Boolean);

c) x or X for prohibited use;

d) n/a, N/A or – (dash) for not applicable;

e) c or C for conditional (see § A.9.2.5).

A.9.2.2 The standardized symbols for the support column are as follows:

a) Y, y, or Yes for implemented;

b) N, n, or No for not implemented.

Space shall be provided for indicating the fact that no support statement is required, in those cases when the status has been evaluated as not applicable. Space shall be provided for reference to a note in those situations where an answer needs a justification or explanation.

A.9.2.3 The conventions given above should be sufficient for the proformas of most protocols. They are not case-sensitive; thus either upper or lower case may be used with the same meaning. If additional conventions are needed, their number should be kept to an absolute minimum, and catalogued by ISO/IEC JTC1/SC21 in order to avoid conflicts with parallel developments.

A.9.2.4 An additional convention can be used for mutually exclusive or selectable options among a set, by placing after the "o" (for optional) a period followed by an integer.

Table A-1/X.291 shows an example of a group of three related options with the meaning that the implementation has to support at least one option in the group of options numbered 4. The PICS proforma shall explicity state, preferably in a footnote to the relevant table, what the requirement is for each numbered group: at least one option is to be supported, or one and only one, or any other requirement.

A.9.2.5 Conditional requirements should be specified in one of the following ways:

a) a "c" is placed in the status column, followed by a colon, followed by one or more unconditional status indications on separate lines, each with a predicate or the negation of a predicate in the predicate column (see § A.9.2.6); negation may be indicated by the operator "^";

D.5.1 *Classes implemented*

Classes implemented				
Item No.	Class	Reference	Status	Support
0	Class 0	14.1	o.1	
1	Class 1	14.2	c1	
2	Class 2	14.3	o.1	
3	Class 3	14.4	c2	
4	Class 4	14.5	c2	

o.1: at least one of these classes shall be supported

c1: IF cls0, THEN o ELSE x

c2: IF cls2, THEN o ELSE x

cls0 = D.5.1/0

cls2 = D.5.1/2

D.6.1 *PDU support*

Supported PDUs						
Item No.	PDU	Reference	Sending		Receipt	
			Status	Support	Status	Support
1	CR	15.1	o		m	
2	CC	15.1	m		c3	
3	DT	15.2	m		m	
. . .	. . .	. . .	. . .		. . .	

c3: IF sendCR, THEN m ELSE n/a

sendCR = D.6.1/1a

D.6.3.1 *Parameters of the XY-PDU*

Supported parameters						
Item No.	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
1	data size	15.6	m		128, 256, 512	
2	timeout	15.7	o		1-3600 s	
3	Class	15.8	m		0-4	
. . .	. . .	. . .	. . .		. . .	

FIGURE A-1/X.291

Examples of PICS proforma tables

TABLE A-1/X.291

Group of related options

Item	Status
Item A	o.4
Item B	o.4
Item C	o.4

o.4: support of at least one of these options is required.

Table A-2/X.291 shows two examples with the following meaning:

1) item A is mandatory if p1 is true, but optional if p1 is false;

2) item B is mandatory if p2 is true, but, by convention, not applicable if p2 is false; there shall be a statement elsewhere in the PICS proforma clarifying this convention, if it is used;

b) a "c" followed by an integer is placed in the status column, providing a reference to a conditional status expression (see § A.9.2.7) defined elsewhere in the PICS proforma, in which case the predicate column may be omitted;

TABLE A-2/X.291

Conditional requirements using predicates

Item	Status	Predicate
Item A	c: m o	p1 ^p1
Item B	c: m	p2

Table A-3/X.291 shows two examples in which the status of each item is determined by evaluating the referenced conditional expression.

Note – A semantically equivalent alternative syntax for conditional requirements may be used, provided that it is catalogued by ISO/IEC JTC1/ SC21.

A.9.2.6 A predicate shall be one of the following:

a) an explicit reference to a PICS proforma Yes/No entry (in the Support column) using the format specified in § A.9.1 b); if the entry is "Yes" then the predicate is True, otherwise it is False; for example, "A.1.2.3/10a" is a predicate which references the first space for an answer in the 10th line of the table found in § A.1.2.3;

b) a predicate name which elsewhere in the PICS proforma is equated with one of the following:

   1) an explicit reference to a PICS proforma Yes/No entry, e.g. "p1", where p1 is defined by a statement:

      "p1 = A.1.2.3/10a"

   2) a relational expression involving a reference to a PICS proforma entry in the Value column, e.g. "p2", where p2 is defined by a statement:

      "p2 = (v2 > 3)"

      where v2 is defined by a statement:

      "v2 = A.1.2.3/10b"

      which references a question that requires an integer answer;

   3) a predicate expression, i.e. a Boolean expression involving predicates, e.g. "p3", where p3 is defined by a statement:

      "p3 = (p1 AND NOT p2) OR (v3 < 2)"

      the syntax and semantics of which should be the same as for Boolean expressions in TTCN (see Recommendation X.292).

TABLE A-3/X.291

References to conditional status expressions

Item	Status
Item A	c1
Item B	c2

A.9.2.7 Conditional status expressions are "IF-THEN-ELSE" expressions which evaluate to an unconditional status depending on the value of the predicate or predicate expression that follows the "IF". If necessary, "IF-THEN-ELSE" expressions may be nested.

For example, c1 and c2 could be defined as follows:

"c1: IF p1 THEN m ELSE o"

"c2: IF (p1 AND NOT p2) OR (v3 < 2) THEN m

ELSE N/A"

Any appropriate syntax may be used for conditional status expressions, but the syntax catalogued by ISO/IEC JTC1/SC21 is recommended in order to avoid an unnecessary multiplicity of syntaxes.

A.9.3 *Instructions for completing a PICS proforma*

The PICS proforma shall contain an additional section:

a) explaining the purpose and the structure of the document to the potential user;

b) explaining the symbols, abbreviations and terms being used, together with appropriate references;

c) giving explicit instructions for completing the PICS;

d) defining the places in which the user can provide additional information.

ANNEX B

(This annex does not form an integral part of this Recommendation)

Guidance for protocol specifiers to facilitate conformance testing

B.1 *Introduction*

This annex gives guidance, primarily for the specifiers of new CCITT Recommendations or International Standards which specify protocols, to facilitate conformance testing by ensuring a very clear understanding of the conformance requirements.

The guidance in this annex on implementation requirements and options should be read in conjunction with the requirements and guidance on PICS proformas in Annex A.

B.2 *Guidance on scope*

B.2.1 Precision in the scope clause sets the tone for precision in the rest of the CCITT Recommendation or International Standard. The requirements stated in the CCITT Recommendation or International Standard should be consistent with the scope and field of application and vice versa.

B.2.2 The scope should distinguish clearly between the following three types of information included in the protocol specification:

a) the definition of the procedures for communication to be followed at the time of communication;

b) requirements to be met by suppliers of implementations of the procedures;

c) guidance on how to implement the procedures.

Guidance on how to implement the procedures does not constitute additional requirements nor does it have any bearing on conformance. If such guidance is included, the scope should make these points and indicate how guidance can be distinguished from the requirements of the specification. This distinction is much easier to make if guidance is separated from requirements. The recommended method of such separation in the ISO/IEC Directives is to place guidance in notes and annexes.

B.2.3 It should be clear to whom the CCITT Recommendation or International Standard applies.

B.2.4 It should be clear under what conditions the CCITT Recommendation or International Standard applies.

Protocol procedures apply between pairs of communicating parties at the time of communication. If there might be any ambiguity over which communicating parties are involved, this should be resolved in the scope.

It is best if protocol specifications are written in such a way that the requirements are to be met by a single communicating party (the “first” communicating party for this purpose) for the benefit of one or more other communicating parties (the “second” communicating parties). Then when two (or more) communicating parties are all expected to communicate in conformance with the specification, the specification is first applied to one party, treating it as the “first” one, and then to the other(s) in turn. This ensures that if the procedures are violated, it is clear which party is at fault.

B.2.5 If any guidance is given about factors which are not standardized definitively, the scope should make it clear that any such guidance can be ignored without affecting conformance.

B.2.6 The aspects which are excluded from the scope should be identified clearly.

Not all factors relevant to the procedures or to products which implement them need to be standardized; indeed it is often desirable to leave some implementor freedom. For instance, it may be desirable to omit in a protocol specification any requirements for explicit values of timeouts, but to give guidance instead.

The scope should make it clear which aspects are standardized definitively, which are covered by guidance but not by any requirements, and which are excluded from consideration by the CCITT Recommendation or International Standard. Any aspects which one might think would be covered, on the basis that they are closely related to aspects which are standardized, need explicit mention.

B.2.7 All options should, if possible, be identified clearly in the scope.

Options are one of the most troublesome, but unfortunately necessary, parts of protocol specifications. They fall somewhere between what is standardized and what is not. They will be covered in greater depth below. What is important is that options are not buried deep inside the specification but are declared clearly at the beginning. If the number and detailed nature of the options makes this impractical, one should seriously ask whether such complexity is really necessary. Can detailed options be grouped together in some way (e.g. classes) to simplify the specification?

B.2.8 The scope clause should be reviewed after considering the rest of the CCITT Recommendation or International Standard.

It is often not possible to satisfy some of the above suggestions until the rest of the CCITT Recommendation or International Standard has been considered. Therefore, it is generally necessary to return to the scope, to check that it really does agree with the contents of the specification. It is common to find that clauses quite outside the scope have been included.

B.3 *Guidance on normative references*

B.3.1 CCITT Recommendations or International Standards which specify OSI protocols should refer to the OSI reference model, the relevant CCITT Recommendations or International Standards which define OSI services and to any relevant CCITT Recommendations or International Standards for appropriate conventions, guidelines, or FDTs.

B.3.2 It should be made clear whether conformance to the CCITT Recommendation or International Standard which specifies the protocol requires conformance to any part of any other CCITT Recommendation or International Standard.

B.3.3 It should be made clear whether a reference is to a particular version of the referenced CCITT Recommendation or International Standard or to each successive version.

Normally, the latest version is required, but this can cause problems as changes to another CCITT Recommendation or International Standard might affect conformance to this one.

B.4 *Guidance on requirements and options*

B.4.1 The status of every requirement should be unambiguous.

Since optional and conditional requirements are so common, there is a tendency to interpret everything which can be interpreted as optional as being optional.

B.4.2 It should be possible for an instance of communication to conform with all the mandatory dynamic conformance requirements.

B.4.3 The conditions under which conditional requirements apply should be spelt out clearly.

B.4.4 It should not be impossible for the implementor or supplier to know what these conditions are.

B.4.5 There should be no possibility of confusion between what is optional dynamically and what is optional statically.

There may be mandatory static conformance requirements for the support of features whose use at the time of communication is optional. Conversely, a message whose use is mandatory in a given context at the time of communication may be part of a protocol mechanism whose support is optional statically.

B.4.6 If the specification contains a "shopping list" of options, and there are restrictions on the allowed combinations of such options, then the restrictions should be specified clearly. These should include identification of any mutual exclusions and any minimum and maximum limits to the allowed range of options.

B.4.7 If the specification does not give any rules for selection of options, it should be made clear in the scope that only the total range and individual options are standardized, but not the selection.

B.4.8 Legitimizing options should be avoided. These are options which allow alternative and incompatible versions of the same thing to claim conformance to the same CCITT Recommendation or International Standard. Although they do not of themselves prevent an objective understanding of conformance, they may frustrate the aims of OSI.

B.4.9 There should be no options which give the implementor permission to ignore important requirements of the specification. Such options devalue the CCITT Recommendation or International Standard and the meaning of conformance to it.

B.4.10 If there are prohibitions in the specification, they should be sufficiently precise to be meaningful.

Many CCITT Recommendations or International Standards have clauses which say in effect "do all of this and nothing else". Such prohibitions may be meaningless, because every protocol conveys some information which is not standardized, the so-called "user data", and every standardized product has attributes which are not standardized, e.g. weight. It may be difficult to draw a clear objective distinction between things the specification cannot forbid and those which the writers of the specification want to forbid, unless the prohibitions are stated explicitly.

B.5 *Guidance on PDUs*

B.5.1 The permitted set of PDU types and parameter encodings should be stated clearly.

B.5.2 The permitted range of values should be stated clearly for each parameter.

B.5.3 All values which fall outside the stated permitted range should be stated explicitly to be invalid.

If not, some people will argue that such values are undefined but allowed, whilst others will argue that they are invalid.

B.5.4 It should be clear whether or not undefined PDU types are allowed.

It is safer to declare all undefined PDU types to be invalid.

B.5.5 Critical undefined values should be mentioned explicitly in the scope as being undefined.

B.5.6 There should be a defined procedure to be followed by the first communicating party in each case of it receiving an invalid or undefined PDU type or parameter.

B.5.7 It should be possible to detect whether the defined procedure has been followed in such cases. If it is not, then it should be because it does not matter.

Sometimes the procedure to be followed upon receipt of an invalid PDU is intentionally the same as when some valid PDUs are received in the same circumstances. For example, the procedure might be to do nothing until one specific type of PDU is received, everything else being ignored. In such situations, it probably does not matter that the error has apparently gone undetected. In other situations, it may be the intention that error cases should be given special treatment, but the procedure has been poorly chosen. In these latter situations, it may be that the action cannot be distinguished from that in the non-error cases.

B.5.8 If, in the encoding of PDUs, there are any fields declared to be "reserved", then there should be a clear statement of what values, if any, are allowed or disallowed in these fields.

B.5.9 If interrelated parameters can be carried on separate PDUs, then the set of permitted relationships between the values of these parameters should be precisely and clearly defined.

B.5.10 If the parameter encoding allows for parameters to be specified in any order, and the PDU format places restrictions on the permitted orders, then these restrictions should be clearly stated. It should be recognized that if many different orders are permitted, then a large representative sample of different orders ought to be tested. The added complexity of testing conformance should, therefore, be adequately compensated by some advantage in allowing this freedom.

B.5.11 The order in which the bits, octets, etc. should be carried in the underlying protocol should be stated clearly.

For example, should a two octet integer travel most or least significant octet first? It is surprising how often such simple causes of ambiguity are overlooked.

B.5.12 The relationship between SDUs and PDUs should be defined clearly.

B.6 *Guidance on states*

B.6.1 Protocol procedures are often defined using a finite state approach, whether formalized or not. The specification of these states is often incomplete.

B.6.2 Each state should be defined clearly.

B.6.3 If there are events which can occur only in a subset of the possible states, then possible occurrence of an event should be distinguished from valid occurrence.

B.6.4 The required actions and state transitions should be defined for each possible state/event pair. In particular, they should be defined for possible but invalid state/event pairs.

B.7 *Guidance on formal description techniques (FDTs)*

B.7.1 The following guidance apply only to those CCITT Recommendations or International Standards which include a formal description. Precise, unambiguous specifications can be written without the aid of an FDT, but in complex CCITT Recommendations or International Standards such as protocols formal descriptions are recommended. It should, however, be realized that they can create problems themselves in relation to conformance.

B.7.2 It should be clear whether the formal description forms a normative part of the International Standard, or integral part of the CCITT Recommendation, which specifies the protocol, or whether it is provided only for guidance.

It is very important to have a clear understanding of the status of the formal description. Ideally there should be no discrepancies between the text and the formal description, but because this is very difficult to achieve in practice it is important that the reader knows which takes precedence. If the formal description is provided only for guidance, it cannot define conformance requirements.

B.7.3 The FDT should be a standardized one and should be properly referenced.

B.7.4 If the formal description defines requirements, but not all the requirements of the CCITT Recommendation or International Standard, then it should be stated clearly that the text includes requirements which are not covered by the formal description and these additional requirements should be identified clearly.

B.7.5 If the formal description defines requirements, and it also defines an allowed way of implementing some aspects of the protocol, but there is intended to be freedom for the implementor to implement those aspects in some other way, then this constitutes over-definition. This is all too common in formal descriptions, and creates difficulties in relation to conformance. If the formal description is an essential part of the CCITT Recommendation or International Standard, then text should be provided to qualify it, indicating where such over-definition exists and what the real requirements are.

The problem usually arises because the formal description describes the internal behaviour of an idealized implementation, rather than the observable external behaviour that is required. It is only the observable external behaviour which can be tested, and therefore it is only this which should constitute requirements for conformance purposes. It may well be that a different FDT should be used for defining the requirements from that used to provide guidance to implementors.

B.8 *Miscellaneous guidance*

Information which may appear obvious should nevertheless be stated.

If something is omitted because it is "obvious", some readers will assume it is required because it is "obvious", but others will assume that it is omitted to provide freedom for implementors. For example, does the existence of a checksum imply that it has to be checked?

ANNEX C

(This annex does not form an integral part of this Recommendation)

Incomplete static conformance requirements

C.1 Some protocol specifications may give an incomplete specification of the static conformance requirements.

C.2 In that case, the PICS proforma should be consulted to clarify what the static conformance requirements are.

C.3 If there is no standardized PICS proforma, then the ATS specifier can accept and state clearly that all capabilities not covered explicitly in the static conformance requirements are optional.

C.4 To minimize the potential problems that this may cause, the ATS specifier can specify that:

a) on reception conforming implementations:

   1) should implement everything that is explicitly specified as mandatory; and

   2) should not omit anything unless it is explicitly stated to be optional, even though there may be a general clause of the sort "if not specified, then optional";

b) on transmission conforming implementations:

   1) should implement everything that is explicitly specified as mandatory; and

   2) may omit anything which is not explicitly stated to be optional, perhaps because of a general clause of the sort "if not specified then optional".

ANNEX D

(This annex does not form an integral part of this Recommendation)

Guidance on generic test cases

D.1 *Introduction*

This annex provides guidance on the production and use of generic test cases specified in TTCN. This does not preclude the specification of other styles of generic test case, if necessary.

D.2 *Description of generic test cases*

A generic test case can consist of a text description of the initial testing state of the test body and a specification of the test body in TTCN. The initial testing state should include not only the protocol state, but also any necessary information concerning the state of the SUT and the testing environment.

The test body

a) should be defined using either the DS or RS test method, in order to avoid the need to specify behaviour of any protocols other than the one which is the focus of the test, and thereby be as test method independent as possible;

b) should assign verdicts in the test body as defined in § 13.2.6.

D.3 *Relation of generic to abstract test cases*

If generic test cases are specified following the guidance of this annex, the main differences between generic and abstract test cases are:

a) the abstract test case includes the specification of a test preamble and a test postamble;

b) the test method used for the test body may be different.

If a generic test suite is produced, it should be used as the means of relating corresponding ATSs for different ATMs.

D.4 *Derivation of abstract test cases from generic test cases*

D.4.1 Once the test method has been chosen, then the generic test cases can be expanded into abstract test cases. There are two main kinds of change required to convert a generic test case into an abstract test case. The first is to express the test body in terms of control and observation required by the test method, and, if relevant include a description of the synchronization needed between upper and lower testers. The second kind of change is to specify the test preamble and postamble.

D.4.2 In converting from generic test cases to abstract test cases, the ATS specifier should ensure that the initial testing state for the test body is preserved, and the sequences of test events defining complete paths through the test body, together with associated verdicts, are preserved.

Printed in Switzerland