



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

CCITT

COMITÉ CONSULTATIF
INTERNATIONAL
TÉLÉGRAPHIQUE ET TÉLÉPHONIQUE

X.290

(11/1988)

SÉRIE X: RÉSEAUX DE COMMUNICATIONS DE
DONNÉES: INTERCONNEXION DE SYSTÈMES
OUVERTS (OSI) – SPÉCIFICATIONS DE PROTOCOLE,
ESSAI DE CONFORMITÉ

Méthodologie des test de conformité

**CADRE ET MÉTHODOLOGIE DES TESTS
DE CONFORMITÉ OSI POUR LES
RECOMMANDATIONS SUR LES
PROTOCOLES POUR LES
APPLICATIONS DU CCITT**

Réédition de la Recommandation du CCITT X.290 publiée
dans le Livre Bleu, Fascicule VIII.5 (1988)

NOTES

1 La Recommandation X.290 du CCITT a été publiée dans le fascicule VIII.5 du Livre Bleu. Ce fichier est un extrait du Livre Bleu. La présentation peut en être légèrement différente, mais le contenu est identique à celui du Livre Bleu et les conditions en matière de droits d'auteur restent inchangées (voir plus loin).

2 Dans la présente Recommandation, le terme «Administration» désigne indifféremment une administration de télécommunication ou une exploitation reconnue.

Recommandation X.290

CADRE ET MÉTHODOLOGIE DES TESTS DE CONFORMITÉ OSI POUR LES RECOMMANDATIONS SUR LES PROTOCOLES POUR LES APPLICATIONS DU CCITT¹⁾

(Melbourne, 1988)

Le CCITT,

considérant

(a) que la Recommandation X.200 définit le modèle de référence pour l'interconnexion de systèmes ouverts pour les applications du CCITT;

(b) que l'objectif de l'OSI ne sera complètement réalisé que si les systèmes dédiés aux applications du CCITT peuvent être testés pour déterminer s'ils sont conformes aux Recommandations de protocole OSI appropriées;

(c) que des suites de tests normalisées doivent être élaborées pour chaque Recommandation de protocole OSI, en sorte:

- que les résultats des tests de conformité provenant de laboratoires de test différents soient acceptés et accueillis avec confiance à un large niveau;
- qu'elles assurent la possibilité d'interfonctionnement des équipements qui ont subi avec succès les tests de conformité normalisés;

(d) la nécessité de définir une Recommandation internationale pour spécifier le cadre et les principes généraux de spécification des suites de tests de conformité et des tests des mises en œuvre des protocoles,

déclare à l'unanimité

(1) que les principes généraux, définitions de termes et de concepts des tests de conformité aux protocoles OSI doivent être conformes à la partie 1 de la présente Recommandation;

(2) que les méthodes, suites et notation des tests doivent être conformes à la partie 2 de la présente Recommandation.

¹⁾ La Recommandation X.290 a été étudiée en étroite collaboration avec les travaux ISO/CEI sur le cadre et la méthodologie des tests de conformité OSI. Au moment de la publication, la Recommandation X.290 a été mise en accord avec les textes de DP 9646/1 et DP 9646/2. Puisque ce travail se trouve à un premier stade de développement, des changements sont attendus. En conséquence, les usagers doivent prendre des précautions pour l'application de la présente Recommandation.

SOMMAIRE

Partie 1 – Concepts généraux

- 0 Introduction
- 1 Objectif et domaine d'application
- 2 Références

SECTION 1 – Terminologie

- 3 Définitions
- 4 Abréviations

SECTION 2 – Présentation générale

- 5 Signification de la conformité dans l'OSI*
- 6 Conformité et tests
- 7 Méthodes de test
- 8 Suites de tests
- 9 Concepts et rôles concernés par chaque partie
- 10 Respect des conditions spécifiées

Partie 2 – Spécification des suites de tests abstraites

- 0 Introduction
- 1 Objectif et domaine d'application
- 2 Références
- 3 Définitions
- 4 Abréviations
- 5 Respect des conditions spécifiées

SECTION 1 – Obligations des concepteurs de protocole

- 6 Conditions de conformité aux normes* OSI*
- 7 Formulaire PICS

SECTION 2 – Obligations des concepteurs de suites de tests abstraites

- 8 Processus de conception des suites de tests
- 9 Détermination des conditions de conformité et des PICS
- 10 Structure des suites de tests
- 11 Spécification des tests élémentaires génériques
- 12 Méthodes de tests abstraites
- 13 Spécifications des suites de tests abstraites
- 14 Utilisation d'une spécification de suite de tests abstraite
- 15 Mise à jour des suites de tests

Annexe A – Options

Annexe B – Directives à l'intention des rédacteurs de Recommandations*

Annexe C – Conditions de conformité statique incomplètes

Annexe D – Notation combinée arborescente et tabulaire (TTCN)

Appendice I – Applicabilité des méthodes de test aux protocoles OSI*

Appendice II – Index des définitions de termes

Appendice III – Exemples de directives concernant des formulaires PICS

Appendice IV – Exemple de choix de méthodes de tests abstraites

Partie 1 – Concepts généraux

0 Introduction

L'objectif de l'OSI ne sera complètement réalisé que quand il sera possible de tester des systèmes pour déterminer leur conformité aux «norme(s) ou Recommandation(s)» (désignées ci-après par «Recommandation(s)*» de protocoles «ISO ou des séries X ou T du CCITT correspondantes» (appelées ci-après OSI*).

Des suites de tests normalisées doivent être élaborées pour chaque Recommandation* de protocole OSI*; elles sont destinées à être utilisées par les fournisseurs et réalisateurs (pour tester eux-mêmes leurs produits), par les utilisateurs de produits OSI, par les Administrations* ou par d'autres organismes de tests indépendants. On devrait parvenir à une comparabilité et à une large acceptation des résultats de tests émanant de différents laboratoires de test, et à réduire par là au minimum la nécessité de répéter des tests de conformité d'un même système.

La normalisation des suites de tests nécessite une définition et une acceptation internationale d'une méthodologie de test commune, ainsi que des procédures et des méthodes de test appropriées. L'objectif de la présente Recommandation est de définir la méthodologie, de fournir un cadre de spécification des suites de tests de conformité et de définir les procédures à appliquer au cours des tests.

Les tests de conformité portent à la fois sur la capacité et sur le comportement d'une réalisation et sur la vérification que ce qui est observé est conforme aux conditions de conformité stipulées dans la ou les Recommandations* concernées, ainsi qu'aux capacités de cette réalisation, telles que déclarées par son réalisateur.

Les tests de conformité ne comportent pas d'évaluation des performances ni de la robustesse ou de la fiabilité d'une réalisation. Ils ne sauraient constituer une appréciation de la réalisation physique des primitives du service abstrait, du mode de réalisation d'un système, de la façon dont il fournit un service demandé, ni de l'environnement de mise en œuvre du protocole. Ils ne peuvent rien prouver, sauf de façon indirecte, sur la conception logique du protocole lui-même.

L'objectif des tests de conformité est d'augmenter la probabilité d'interfonctionnement de réalisations différentes. A cette fin, ces réalisations sont soumises à une suite de tests de protocole, de manière à donner une meilleure confiance en la conformité de chaque réalisation à sa spécification de protocole. La confiance en la conformité à une spécification de protocole est particulièrement importante quand l'interfonctionnement d'équipements de fournisseurs différents est impératif.

Il convient toutefois de garder à l'esprit que la complexité de la plupart des protocoles rend les tests exhaustifs impraticables, sur le plan technique comme sur le plan économique. En outre, les tests ne sauraient garantir la conformité à une spécification, puisqu'ils détectent les erreurs, et non leur absence. Ainsi, la conformité prouvée par une suite de tests ne peut pas, à elle seule, garantir l'interfonctionnement. Mais elle donne l'assurance qu'une réalisation a les capacités requises et que son comportement dans des instances de communication représentatives est conforme et cohérent.

A noter que le modèle de référence OSI pour les applications du CCITT (Recommandation X.200) stipule (au § 4.3):

«Seul le comportement extérieur des systèmes ouverts est retenu pour la définition des normes des systèmes ouverts réels.»

Ceci signifie que, même si les Recommandations* OSI* décrivent des aspects des comportements intérieur et extérieur, seules les conditions concernant le comportement extérieur doivent être satisfaites par des systèmes ouverts réels. Bien que certaines des méthodes définies dans la présente Recommandation imposent certaines contraintes au concepteur, par exemple l'existence de moyens permettant de réaliser un contrôle et une observation à un ou plusieurs points d'accès à des services, il faut signaler que d'autres méthodes, également définies ici, n'imposent pas de telles contraintes.

Toutefois, dans le cas de systèmes d'extrémité OSI* partiels, qui mettent en œuvre des protocoles OSI* jusqu'à la frontière d'une couche spécifique, il est désirable que les tests portent sur le comportement extérieur des entités de protocole mises en œuvre et sur la capacité de ces entités à permettre un comportement extérieur correct dans les couches de niveaux supérieurs.

Une analyse détaillée des avantages relatifs et de l'efficacité des diverses méthodes, ainsi que des contraintes qu'elles imposent, est présentée dans diverses parties de la Recommandation. Toutefois, toute organisation envisageant l'utilisation des méthodes de test définies dans la présente Recommandation dans un contexte tel qu'une certification, doit examiner soigneusement les contraintes relatives à l'applicabilité et les avantages des diverses méthodes de test possibles.

En ce qui concerne l'ISO et le CCITT, les tests sont volontaires. Les spécifications de tests de recette, ou concernant d'autres procédures contractuelles extérieures, ne sont pas concernées par la normalisation.

1 Objectif et domaine d'application

1.1 La présente Recommandation spécifie une méthodologie générale de tests de conformité à une ou à des Recommandations* de protocole OSI* pour les produits déclarés comme mettant en œuvre la ou lesdites Recommandations*. La méthodologie concerne également les tests de conformité à une ou des Recommandations de syntaxe de transfert, dans la mesure où cette conformité peut être déterminée par des tests portant sur la combinaison de chacune d'elles avec un protocole OSI spécifique.

1.2 La présente Recommandation est structurée en deux parties distinctes:

La partie 1 recense les différentes phases du processus de test de conformité, ces phases étant caractérisées par quatre rôles principaux. Ces rôles sont:

- a) la spécification de suites de tests abstraites pour des protocoles OSI* particuliers;
- b) la dérivation de suites de tests exécutables et les outils de test associés;
- c) le rôle du client d'un laboratoire de test, qui veut tester une réalisation mettant en œuvre des protocoles OSI*;
- d) l'exécution des tests de conformité, dont le point essentiel est l'établissement d'un rapport de test de conformité, qui donne les résultats selon les termes de la ou des Recommandations* et des suites de tests utilisées.

En outre, cette partie donne des informations de nature didactique, ainsi que les définitions de concepts et de termes.

La partie 2 définit les obligations et directives pour la spécification de suites de tests abstraites pour les protocoles OSI*.

1.3 Dans les deux parties de la présente Recommandation, le domaine est limité aux seules informations nécessaires à répondre aux objectifs suivants:

- a) atteindre un niveau de confiance adéquat dans les tests de conformité;
- b) parvenir à une comparabilité des résultats de tests correspondants, réalisés sur des sites différents et à des moments différents;
- c) faciliter la communication entre les partenaires remplissant les rôles décrits ci-dessus.

1.4 Un autre aspect de cet objectif est le cadre d'élaboration de suites de tests OSI*. Par exemple:

- a) comment elles peuvent concerner les divers types de conditions de conformité;
- b) les types de tests à normaliser et ceux ne nécessitant pas de normalisation;
- c) les critères de sélection des tests à inclure dans une suite de tests de conformité;
- d) la notation à utiliser pour la définition de tests;
- e) la structure d'une suite de tests.

1.5 La certification, procédure administrative qui peut suivre des tests de conformité, n'entre pas dans le cadre de la présente Recommandation.

1.6 Les protocoles de la couche physique et de contrôle de l'accès au support n'entrent pas dans le cadre de la présente Recommandation.

2 Références

Recommandation X.200 – *Modèle de référence pour l'interconnexion des systèmes ouverts pour les applications du CCITT* (voir également ISO 7498).

Recommandation X.210 – *Conventions relatives à la définition de service des couches de l'interconnexion de systèmes ouverts* (voir également ISO TR 8509).

Recommandation X.209 – *Spécifications des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1)* (voir également ISO 8825).

3 Définitions

3.1 Définitions du modèle de référence

La présente Recommandation est fondée sur les concepts développés dans le modèle de référence pour l'interconnexion des systèmes ouverts pour les applications du CCITT (Recommandation X.200) et utilise les termes suivants qui y sont définis:

- a) entité (N)
- b) service (N)
- c) couche (N)
- d) protocole (N)
- e) point d'accès aux services (N)
- f) relais (N)
- g) unité de données du protocole (N)
- h) informations de contrôle du protocole (N)
- i) données de l'utilisateur (N)
- j) système ouvert réel
- k) sous-réseau
- l) entité d'application
- m) élément de service d'application
- n) syntaxe de transfert
- o) couche physique
- p) couche liaison de données
- q) couche réseau
- r) couche transport
- s) couche session
- t) couche présentation
- u) couche application
- v) gestion du système
- w) gestion d'application
- x) gestion de couche.

3.2 Termes définis dans d'autres Recommandations

La présente Recommandation utilise les termes suivants définis dans les Conventions de service OSI (Recommandation X.210):

- a) utilisateur du service
- b) fournisseur du service.

La présente Recommandation utilise le terme suivant défini dans la spécification des règles de codage de base ASN.1 (Recommandation X.209):

- c) codage.

3.3 Définitions relatives aux tests de conformité

Aux fins de la présente Recommandation, les définitions des § 3.4 à 3.8 sont également applicables.

3.4 *Termes de base*

3.4.1 **réalisation à tester (IUT)**

Partie d'un système ouvert réel qui doit être soumise à des tests et qui peut mettre en œuvre un ou plusieurs protocoles OSI* dans une relation avec un fournisseur ou un utilisateur adjacent.

3.4.2 **système à tester (SUT)**

Le système ouvert réel où réside l'IUT.

3.4.3 **conditions de conformité dynamique**

Toutes les conditions (et options) qui déterminent le comportement observable autorisé par la ou les Recommandations* OSI* dans des instances de communication.

3.4.4 **conditions de conformité statique**

Contrainte formulée dans des Recommandations* OSI* pour faciliter l'interfonctionnement, en définissant des spécifications relatives aux possibilités d'une réalisation.

Remarque – Les conditions de conformité statique peuvent être d'un niveau général, comme le regroupement d'unités fonctionnelles et les options de classes de protocole, ou d'un niveau détaillé, comme les plages de valeurs que doivent pouvoir prendre des paramètres spécifiques ou des délais de temporisation.

3.4.5 **capacités d'une IUT**

Ensemble des fonctions et options du ou des protocoles concernés et, le cas échéant, ensemble des facilités et options formulées dans la définition de service concernée, qui doivent être assurées par cette IUT.

3.4.6 **déclaration de conformité d'une mise en œuvre de protocole (PICS)**

Déclaration faite par le fournisseur d'une réalisation ou d'un système OSI, précisant les capacités et options qui ont été réalisées et les fonctions qui n'ont pas été réalisées.

3.4.7 **formulaire PICS**

Document, sous forme de questionnaire, établi par le concepteur du protocole ou de la suite de tests de conformité; rempli pour une réalisation ou un système OSI* il devient la PICS.

3.4.8 **informations complémentaires sur la mise en œuvre du protocole destinées au test (PIXIT)**

Déclaration faite par le fournisseur ou le réalisateur d'une IUT, contenant toutes les informations, en plus de celles fournies dans la PICS, concernant l'IUT et son environnement de test, ou y renvoyant; elle permettra au laboratoire de test d'exécuter sur cette IUT la suite de tests appropriée.

3.4.9 **formulaire PIXIT**

Document, sous forme de questionnaire, fourni par le laboratoire de test et destiné à être rempli lors de la préparation du test; il devient alors le PIXIT.

3.4.10 **réalisation conforme**

IUT dont il est démontré qu'elle satisfait les conditions de conformité statique et dynamique, en accord avec les capacités déclarées dans la PICS.

3.4.11 **déclaration de conformité d'un système**

Document précisant les Recommandations* OSI* mises en œuvre et auxquelles il est déclaré conforme.

3.4.12 **client**

Organisme qui soumet un système ou une réalisation à des tests de conformité.

3.4.13 **laboratoire de test**

Organisation qui exécute des tests de conformité. Ce peut être un organisme indépendant, un groupe d'utilisateurs, une Administration* ou une structure identifiable faisant partie de l'organisme fournisseur.

3.5 *Termes relatifs aux types de tests*

3.5.1 **tests actifs**

Application, dans des conditions contrôlées, d'une suite de tests à un SUT, afin d'observer les réactions de l'IUT.

3.5.2 tests passifs

Observation de l'activité des PDU sur une liaison de communication, pour déterminer si le comportement observé est autorisé ou non par la ou les Recommandations* concernées.

3.5.3 test multicouche

Test du comportement d'une IUT multicouche effectué globalement et non couche par couche.

3.5.4 test encastré

Test du comportement d'une seule couche d'une IUT multicouche, réalisé sans accéder aux frontières de cette couche.

3.5.5 test d'interconnexion de base

Test limité d'une IUT, visant à déterminer, sans se lancer dans un test complet, si elle est suffisamment conforme aux caractéristiques principales du ou des protocoles concernés pour que l'interconnexion soit possible.

3.5.6 test de capacités

Test de détermination des capacités d'une IUT.

Remarque – Ce test comprend la vérification de toutes les capacités obligatoires, ainsi que des capacités optionnelles déclarées offertes dans la PICS, mais pas celle des capacités déclarées non offertes dans la PICS.

3.5.7 revue de conformité statique

Revue visant à déterminer jusqu'à quel point les conditions de conformité statique sont remplies par l'IUT, et consistant en une comparaison de la PICS et des résultats de tous les tests de capacités portant sur les conditions de conformité statique exprimées dans la ou les Recommandations* concernées.

3.5.8 test de comportement

Test visant à déterminer jusqu'à quel point les conditions de conformité dynamique sont remplies par l'IUT.

3.5.9 test de conformité

Test visant à déterminer jusqu'à quel point l'IUT est une réalisation conforme.

3.5.10 évaluation de conformité

Exécution complète de toutes les activités de test nécessaires à l'évaluation de la conformité d'une réalisation ou d'un système à une ou plusieurs Recommandations* OSI*. Elle comprend l'établissement des documents PICS et PIXIT, la préparation du testeur réel et du SUT, l'exécution d'une ou plusieurs suites de tests, l'analyse des résultats et l'établissement des rapports de tests de conformité propres au système (SCTR) et à chacun des protocoles (PCTR).

3.6 Termes relatifs aux suites de tests

3.6.1 méthode de test abstraite

Description de la façon dont une IUT doit être testée, formulée à un niveau suffisamment abstrait pour être indépendante de toute réalisation particulière d'outils de test, mais suffisamment détaillé pour permettre la spécification des tests.

3.6.2 méthodologie de test abstraite

Méthode de description et de classification des méthodes de test abstraites.

3.6.3 test élémentaire abstrait

Spécification complète et indépendante des actions nécessaires afin de réaliser l'objet d'un test spécifique, définies au niveau d'abstraction d'une méthode de test abstraite particulière. Il comprend un préambule et un épilogue, garantissant son lancement et sa terminaison dans un état stable (c'est-à-dire un état pouvant être maintenu presque indéfiniment, comme l'état «repos» ou «transfert de données») et met en jeu une ou plusieurs connexions consécutives ou concurrentes.

Remarque 1 – La spécification doit être complète, en ce sens qu'elle doit être suffisante pour permettre de rendre un verdict non ambigu sur chaque résultat potentiellement observable (c'est-à-dire à l'issue de chaque séquence d'événements de test).

Remarque 2 – La spécification doit être indépendante, en ce sens qu'il doit être possible d'exécuter le test élémentaire exécutable dérivé, indépendamment des autres tests élémentaires de cette nature (c'est-à-dire que la spécification doit toujours impliquer la possibilité de lancement et de terminaison dans l'état «repos» – dans lequel il

n'existe aucune connexion en dehors des connexions permanentes). Pour certains tests élémentaires, des conditions préalables peuvent être formulées, en ce sens que leur exécution nécessite certaines capacités spécifiques de l'IUT, qui ont été confirmées par les résultats de tests élémentaires exécutés antérieurement.

3.6.4 **test élémentaire exécutable**

Réalisation d'un test élémentaire.

Remarque – En général, l'utilisation du mot «test» implique le sens normal de ce terme. Il peut quelque fois être utilisé comme abréviation de «test élémentaire abstrait» ou de «test élémentaire exécutable». Le contexte devrait déterminer le sens exact.

3.6.5 **objet du test**

Description du but que le test élémentaire abstrait est censé atteindre.

3.6.6 **test élémentaire générique**

Spécification des actions nécessaires afin de remplir l'objet d'un test spécifique, définies par un corps de test et une description de l'état initial à partir duquel le corps du test doit être lancé.

3.6.7 **préambule**

Modules de test nécessaires afin de définir le passage de l'état stable de lancement du test élémentaire à l'état initial de lancement du corps du test.

3.6.8 **corps du test**

Ensemble des modules de test qui sont essentiels pour remplir l'objet du test et rendre des verdicts sur les résultats possibles.

3.6.9 **épilogue**

Modules de test nécessaires afin de définir le passage de la fin du corps du test à l'état stable terminant le test élémentaire.

3.6.10 **module de test**

Subdivision d'un test élémentaire, dotée d'un nom, composée d'événements de test ou d'autres tests élémentaires, et utilisée pour structurer des tests élémentaires abstraits.

3.6.11 **événement de test**

Unité de spécification de test, indivisible au niveau d'abstraction de cette spécification (par exemple, expédition ou réception d'une seule PDU).

3.6.12 **suite de tests**

L'ensemble complet de tests élémentaires, éventuellement combinés en groupes de tests imbriqués, nécessaires à effectuer le test de conformité d'une IUT, le test d'interconnexion de base d'une IUT, ou le test de mise en œuvre d'un protocole dans une IUT.

3.6.13 **test élémentaire**

Test élémentaire générique, abstrait ou exécutable.

3.6.14 **groupe de tests**

Ensemble, doté d'un nom, de tests élémentaires associés.

3.6.15 **suite de tests génériques**

Suite de tests composée de tests élémentaires génériques, ayant la même portée que l'ensemble complet des objets de test d'un protocole particulier, celui-ci étant l'ensemble, ou un sur-ensemble, des objets de test de toute suite de tests abstraite de ce protocole.

3.6.16 **suite de tests abstraite**

Suite de tests composée de tests élémentaires abstraits.

3.6.17 **suite de tests exécutables**

Suite de tests composée de tests élémentaires exécutables.

3.6.18 **suite de tests de conformité**

Suite de tests de conformité portant sur un ou plusieurs protocoles OSI*.

Remarque – Cette suite doit comprendre les tests de capacité et les tests de comportement. Elle peut être qualifiée par les adjectifs: abstraite, générique ou exécutable, suivant le cas. Sauf spécification contraire, elle est supposée être une «suite de tests abstraite».

3.6.19 **suite de tests d'interconnexion de base**

Suite de tests portant sur une interconnexion de base mettant en œuvre un ou plusieurs protocoles OSI.

3.6.20 **suite de tests abstraite sélectionnée**

Sous-ensemble d'une suite de tests abstraite, sélectionné d'après une PICS spécifique.

3.6.21 **suite de tests exécutable sélectionnée**

Sous-ensemble d'une suite de tests exécutable, sélectionné d'après une PICS spécifique et correspondant à une suite de tests abstraite sélectionnée.

3.6.22 **test élémentaire abstrait paramétré**

Test élémentaire abstrait, dont tous les paramètres appropriés ont été fixés à des valeurs conformes à une PICS et à un PIXIT spécifiques.

3.6.23 **test élémentaire exécutable paramétré**

Test élémentaire exécutable, dont tous les paramètres appropriés ont été fixés à des valeurs conformes à une PICS et à un PIXIT spécifiques.

3.6.24 **suite de tests abstraite paramétrée**

Suite de tests abstraite sélectionnée dont tous les tests élémentaires ont été paramétrés conformément à la PICS et au PIXIT appropriés.

3.6.25 **suite de tests exécutable paramétrée**

Suite de tests exécutable sélectionnée dont tous les tests élémentaires ont été paramétrés conformément à la PICS et au PIXIT appropriés, et correspondant à une suite de tests abstraite.

3.7 *Termes relatifs aux résultats*

3.7.1 **reproductibilité (des résultats)**

Caractéristique d'un test élémentaire, d'après laquelle des exécutions répétées sur la même IUT conduisent au même verdict; par extension, caractéristique d'une suite de tests.

3.7.2 **comparabilité (des résultats)**

Caractéristique d'évaluations de conformité, d'après laquelle leurs exécutions sur la même IUT, dans des environnements de test différents, conduisent au même résultat global.

3.7.3 **résultats**

Séquence d'événements de tests, avec leurs entrées/sorties respectives, identifiée par un concepteur de test abstrait, ou observée durant l'exécution du test.

3.7.4 **résultat prévu**

Résultat identifié ou répertorié dans la spécification d'un test élémentaire abstrait.

3.7.5 **résultat imprévu**

Résultat non identifié ou répertorié dans la spécification d'un test élémentaire abstrait.

3.7.6 **verdict**

Jugement «succès», «échec» ou «non concluant» rendu sur la conformité d'une IUT, à l'issue d'un test élémentaire qui est spécifié dans la suite de tests abstraite.

3.7.7 **rapport de test de conformité du système (SCTR)**

Document rédigé à la fin de l'évaluation de conformité, fournissant un résumé général sur la conformité du système à l'ensemble des protocoles pour lesquels les tests de conformité ont été effectués.

3.7.8 **rapport de test de conformité au protocole (PCTR)**

Document établi à la fin de l'évaluation de conformité, donnant le détail des tests effectués pour un protocole particulier, et comprenant l'identification des tests élémentaires abstraits dont les tests élémentaires exécutables correspondants ont été exécutés et, pour chacun de ces tests élémentaires, l'objet du test et le verdict.

3.7.9 **événement de test valide**

Événement de test autorisé par la Recommandation* de protocole, syntaxiquement correct, survenant alors qu'il est attendu, et se présentant dans un contexte autorisé d'un résultat observé.

3.7.10 **événement de test syntaxiquement non valide**

Événement de test qui n'est pas syntaxiquement autorisé par la Recommandation* de protocole.

Remarque – L'utilisation de l'expression «événement de test non valide» est déconseillée.

3.7.11 **événement de test inopportun**

Événement de test qui, bien que syntaxiquement correct, survient alors qu'il n'est pas autorisé par la Recommandation* de protocole, ou se présente en un point d'un résultat observé auquel, d'après cette Recommandation*, il ne devrait pas se présenter.

3.7.12 **verdict «succès»**

Verdict rendu quand les résultats observés satisfont l'objet du test et sont valides au regard de la ou des Recommandations* concernées et de la PICS.

3.7.13 **verdict «échec»**

Verdict rendu quand le résultat observé est syntaxiquement non valide ou inopportun au regard de la ou des Recommandations* concernées ou de la PICS.

3.7.14 **verdict «non concluant»**

Verdict rendu quand le résultat observé est valide au regard de la ou des Recommandations* concernées, mais ne permet pas de remplir l'objet du test.

3.7.15 **journal de conformité**

Enregistrement des informations nécessaires à la vérification du verdict rendu issu des tests de conformité.

3.8 *Termes relatifs aux méthodes de test*

3.8.1 **point de contrôle et d'observation (PCO)**

Point auquel l'exercice d'un contrôle et d'une observation est spécifié dans un test élémentaire.

3.8.2 **testeur inférieur**

Formulation abstraite des moyens d'exercer, au cours de l'exécution du test, le contrôle et l'observation au PCO approprié, situé en dessous ou distant de l'IUT, comme spécifié par la méthode de test abstraite choisie.

3.8.3 **testeur supérieur**

Formulation abstraite des moyens d'exercer, au cours de l'exécution du test, le contrôle et l'observation de la frontière de service supérieure de l'IUT, ainsi que de toutes primitives locales abstraites concernées.

3.8.4 **primitive abstraite du service (N) [ASP(N)]**

Description indépendante de la réalisation, d'une interaction entre un utilisateur et un fournisseur du service, au niveau d'une frontière du service (N), telle que définie dans une Recommandation* de service OSI*.

3.8.5 **primitive locale abstraite (ALP)**

Abréviation utilisée pour décrire un contrôle et/ou une observation que doit exercer le testeur supérieur, qui ne peut pas être décrite sur la base d'ASP, mais qui concerne des événements ou des états définis dans la ou les Recommandations* de protocole concernant l'IUT.

Remarque – Le PIXIT doit indiquer si une ALP particulière peut ou non être réalisée dans le SUT. L'aptitude du SUT à prendre en charge une ALP particulière, comme spécifié dans le PIXIT, sera un critère de sélection des tests.

3.8.6 **procédures de coordination de tests**

Règles de coopération entre les testeurs inférieur et supérieur durant des tests.

3.8.7 **protocole de gestion des tests**

Protocole utilisé comme réalisation des procédures de coordination des tests pour une suite particulière de tests.

3.8.8 **méthode de test locale**

Méthode de test abstraite dont les PCO sont situés directement aux frontières de la couche de l'IUT.

3.8.9 **méthode de test externe**

Méthode de test abstraite dans laquelle le testeur inférieur est séparé du SUT et communique avec lui par l'intermédiaire d'un fournisseur du service approprié de la couche de niveau inférieur.

Remarque – Le fournisseur du service est situé immédiatement en dessous du protocole (de la couche la plus basse de l'IUT) sur lequel est axé le test, et peut faire intervenir plusieurs couches OSI.

3.8.10 **méthode de test répartie**

Méthode de test externe impliquant un PCO à la frontière de la couche la plus élevée de l'IUT.

3.8.11 **méthode de test coordonnée**

Méthode de test externe pour laquelle un protocole de gestion de tests est défini comme réalisation des procédures de coordination de tests, et permettant de spécifier le contrôle et l'observation uniquement en fonction de l'activité du testeur inférieur, incluant le contrôle et l'observation des PDU de gestion de test.

3.8.12 **méthode de test à distance**

Méthode externe qui ne comporte ni de PCO au-dessus de l'IUT ni de protocole de gestion de tests normalisé; certaines spécifications de procédures de coordination de tests peuvent être implicites, formulées de façon informelle dans la suite de tests abstraite, mais sans rien supposer quant à leur faisabilité ou leur réalisation.

3.8.13 **testeur réel**

Réalisation du testeur inférieur, complétée par la définition ou la réalisation du testeur supérieur, et de la définition des procédures de coordination de tests – le tout propre à une méthode de test particulière.

3.8.14 **réalisateur de l'équipement de test**

Organisme qui se charge de fournir, sous une forme indépendante du client et de l'IUT, les moyens de tester la conformité de l'IUT selon une suite de tests abstraite.

4 **Abréviations**

Les abréviations suivantes sont utilisées dans la présente Recommandation:

Administration*	Administration ou organisme d'exploitation privé reconnu
ALP	primitive locale abstraite
APS	primitive de service abstraite
ETTD	équipement terminal de traitement de données
IUT	réalisation à tester
OSI	interconnexion des systèmes ouverts
OSI*	OSI ou les Recommandations correspondantes des séries X ou T du CCITT
PCO	point de contrôle et d'observation
PCTR	rapport de test de conformité au protocole
PDU	unité de données de protocole
PICS	déclaration de conformité d'une mise en œuvre de protocole
PIXIT	informations complémentaires sur la mise en œuvre du protocole destinées aux tests
SAP	point d'accès aux services
SCTR	rapport de test de conformité du système
Recommandation*	norme ou Recommandation
SUT	système à tester
TMPDU	PDU de gestion de tests

5 Signification de la conformité dans l'OSI*

5.1 Introduction

Dans le contexte de l'OSI*, un système réel est dit conforme s'il satisfait les Recommandations OSI* applicables, dans sa communication avec d'autres systèmes réels.

Les Recommandations* OSI* applicables comprennent les Recommandations de protocole et les Recommandations de syntaxe de transfert, dans la mesure où elles sont mises en œuvre en conjonction avec les protocoles.

Les Recommandations* OSI forment un ensemble de Recommandations* en relation, qui définissent le comportement des systèmes ouverts dans leur communication. La conformité d'un système réel sera donc exprimée à deux niveaux: la conformité à chaque Recommandation* et la conformité à l'ensemble des Recommandations*.

Remarque – Si la réalisation est fondée sur un ensemble prédéfini de Recommandations*, souvent appelé norme ou profil fonctionnels, le concept de conformité peut être étendu aux conditions spécifiques exprimées dans cette norme ou ce profil fonctionnels, à condition qu'il n'entre pas en conflit avec les conditions spécifiées dans les Recommandations* de base.

5.2 Conditions de conformité

5.2.1 Les conditions de conformité d'une Recommandation* peuvent être:

- a) des conditions obligatoires: elles doivent être observées dans tous les cas;
- b) des conditions conditionnelles: elles doivent être observées si les conditions fixées dans la Recommandation* s'appliquent;
- c) des options: elles peuvent être choisies pour s'adapter à la réalisation, à condition que toutes les conditions applicables à l'option soient respectées. De plus amples informations sur les options sont données dans l'annexe A.

Par exemple, les éléments de service de base du CCITT sont des conditions obligatoires; les éléments de service optionnels peuvent être des conditions conditionnelles ou optionnelles.

Remarque – Les expressions du CCITT «élément de service de base» et «élément de service optionnel» doivent être considérées dans le contexte du domaine d'application des Recommandations du CCITT concernées; dans beaucoup de cas, des éléments de service de base sont obligatoires pour des réseaux mais pas pour des ETDD.

5.2.2 En outre, les conditions de conformité formulées dans une Recommandation* peuvent être énoncées:

- a) positivement: elles déclarent ce qui doit être fait;
- b) négativement (interdictions): elles déclarent ce qui ne doit pas être fait.

5.2.3 Enfin, les conditions de conformité se répartissent en deux groupes:

- a) les conditions de conformité statique;
- b) les conditions de conformité dynamique,

respectivement traitées aux § 5.3 et 5.5.

5.3 Conditions de conformité statique

Les conditions de conformité statique sont celles qui définissent les capacités autorisées minimales d'une réalisation devant exister pour faciliter son interfonctionnement. Ces capacités peuvent être d'un niveau général, comme le regroupement d'unités fonctionnelles et d'options en classes de protocole, ou d'un niveau détaillé, comme la plage des valeurs que doivent pouvoir prendre des paramètres ou des délais de temporisation spécifiques.

Les conditions et options de conformité statique des Recommandations* OSI* peuvent être de deux ordres:

- a) celles qui déterminent les capacités à inclure dans une réalisation mettant en oeuvre un protocole particulier;
- b) celles qui déterminent des dépendances multicouches, c'est-à-dire qui imposent des contraintes aux capacités des couches sous-jacentes au système où est réalisée la mise en oeuvre du protocole. Elles se rencontrent plutôt dans les Recommandations* concernant les couches des niveaux supérieurs.

Toutes les capacités non explicitement déclarées comme conditions de conformité statique doivent être considérées comme optionnelles.

5.4 *Déclaration de conformité d'une mise en oeuvre de protocole (PICS)*

Pour évaluer la conformité d'une réalisation particulière, il est nécessaire de disposer d'une déclaration des capacités et options mises en oeuvre et de toutes les caractéristiques qui ont été omises, en sorte de pouvoir tester la conformité de la réalisation aux conditions concernées et à celles-là seulement. Une telle déclaration est appelée déclaration de conformité d'une mise en oeuvre de protocole (PICS).

Il convient de distinguer les catégories d'informations suivantes, que peut contenir une PICS:

- a) les informations concernant les conditions de conformité statique obligatoires, optionnelles et conditionnelles du protocole proprement dit;
- b) les informations concernant les conditions de conformité statique obligatoires, optionnelles et conditionnelles concernant les dépendances multicouches.

Si un ensemble de Recommandations de protocole OSI en relation a été mis en oeuvre dans un système, une PICS est nécessaire pour chaque protocole. Une déclaration de conformité du système est également nécessaire. Elle indique tous les protocoles du système pour lesquels une PICS distincte est fournie.

5.5 *Conditions de conformité dynamique*

Les conditions de conformité dynamique (et leurs options) sont celles qui déterminent le comportement observable autorisé par la ou les Recommandations* OSI* dans des instances de communication. Elles constituent le corps de chaque Recommandation* de protocole OSI*. Ces conditions définissent l'ensemble des comportements autorisés d'une réalisation ou d'un système réel. Cet ensemble définit les capacités maximales qu'une réalisation ou un système réel conforme peut présenter, selon les termes de la Recommandation* de protocole OSI*.

Un système présente une conformité dynamique dans une instance de communication si son comportement est un membre de l'ensemble de tous les comportements autorisés par la ou les Recommandations* de protocole OSI* concernées, d'une façon conforme à la PICS.

5.6 *Système conforme*

Un système ou une réalisation est dit conforme s'il satisfait les conditions de conformité statique et dynamique, selon les capacités déclarées dans la PICS, pour chaque protocole stipulé dans la déclaration de conformité du système.

5.7 *Interfonctionnement et conformité*

5.7.1 L'objet essentiel des tests de conformité est d'augmenter la probabilité d'interfonctionnement de réalisations différentes.

L'interfonctionnement de deux systèmes ouverts réels, ou plus, a davantage de chances de réussir s'ils se conforment tous au même sous-ensemble d'une Recommandation* OSI* ou à la même sélection de Recommandations* OSI*, que dans le cas contraire.

Pour préparer deux ou plusieurs systèmes à un interfonctionnement réussi, il est recommandé de comparer les déclarations de conformité de ces systèmes ainsi que leurs PICS.

Si plusieurs versions d'une Recommandation* OSI* concernée sont indiquées dans la PICS, il faut déterminer les différences entre ces versions et tenir compte de leurs conséquences, qui peuvent inclure leur utilisation en combinaison avec d'autres Recommandations*.

5.7.2 Bien que la conformité soit une condition nécessaire, elle n'est pas en elle-même une condition suffisante pour garantir la possibilité d'interfonctionnement. Même si deux réalisations sont conformes à la même Recommandation* de protocole OSI*, leur interfonctionnement peut échouer à cause de facteurs qui n'entrent pas dans le cadre de la présente Recommandation.

Pour détecter ces facteurs, un essai d'interfonctionnement est recommandé. Des informations complémentaires pouvant aider à réaliser l'interfonctionnement de deux systèmes, peuvent être obtenues en étendant la comparaison des PICS à celle d'autres informations pertinentes, comprenant les rapports de test et les PIXIT (voir le § 6.2). La comparaison peut être axée sur:

- a) des mécanismes additionnels déclarés comme palliant des ambiguïtés ou des déficiences connues, qui n'ont pas encore été corrigées dans les Recommandations* ou dans des systèmes réels homologues, par exemple des solutions à des problèmes multicouches;
- b) une sélection d'options libres qui n'est pas prise en compte dans les conditions de conformité statique des Recommandations*;
- c) l'existence de temporisateurs non spécifiés dans les Recommandations* et les délais associés.

Remarque – La comparaison peut porter sur deux systèmes individuels, deux ou plusieurs types de produits ou, pour la comparaison de PICS uniquement, deux ou plusieurs spécifications de recette, autorisation de connexion, etc.

6 Conformité et tests

6.1 Objectifs des tests de conformité

6.1.1 Introduction

La présente Recommandation traite principalement des tests de conformité aux Recommandations* de protocole OSI*. Mais elle concerne également les tests de conformité aux Recommandations* de syntaxe de transfert OSI* dans la mesure où ils peuvent être effectués en testant la syntaxe de transfert en combinaison avec un protocole OSI*.

En principe, l'objectif des tests de conformité est de déterminer si la réalisation testée est conforme à la spécification de la Recommandation* pertinente. Des limitations pratiques rendent impossible l'exhaustivité des tests, auxquels d'autres restrictions peuvent s'appliquer pour des considérations économiques.

La présente Recommandation distingue donc quatre types de test, selon la portée de l'indication de conformité qu'ils fournissent:

- a) les tests d'interconnexion de base, qui fournissent une indication à *première vue* de la conformité d'une IUT;
- b) les tests de capacités, qui vérifient que les capacités observables de l'IUT sont conformes aux conditions de conformité statique et aux capacités déclarées dans la PICS;
- c) les tests de comportement, qui visent à être aussi complets que possible, sur toute la gamme des conditions de conformité dynamique spécifiées par la Recommandation* dans la limite des capacités de l'IUT;
- d) les tests de résolution de conformité, qui vérifient à fond la conformité d'une IUT à des conditions particulières, en sorte de fournir des réponses claires, positives ou négatives et des informations de diagnostic concernant des aspects spécifiques de la conformité; ces tests ne sont pas normalisés.

6.1.2 Tests d'interconnexion de base

6.1.2.1 Les tests d'interconnexion de base d'une IUT sont limités aux principales caractéristiques énoncées dans une Recommandation*, et visent à établir, sans entreprendre un test complet, si la conformité est suffisante pour permettre l'interconnexion.

6.1.2.2 Les tests d'interconnexion de base sont appropriés:

- a) pour détecter les cas graves de non-conformité;
- b) comme filtre préliminaire avant d'entreprendre des tests plus coûteux;
- c) pour donner une indication à *première vue* qu'une réalisation qui a subi avec succès l'intégralité des tests dans un certain environnement sera encore conforme dans un nouvel environnement [par exemple, avant de tester une réalisation (N), vérifier qu'une réalisation (N – 1) n'a pas été modifiée de façon importante du fait de sa liaison avec cette réalisation (N)];
- d) pour utilisation par les utilisateurs des réalisations, pour déterminer si ces réalisations paraissent utilisables pour la communication avec d'autres réalisations conformes, par exemple comme préliminaire à un transfert de données.

6.1.2.3 Les tests d'interconnexion de base ne sont pas appropriés:

- a) comme base de déclaration de conformité du fournisseur d'une réalisation;
- b) pour déterminer les causes d'un échec de communication.

6.1.2.4 Les tests d'interconnexion de base doivent être normalisés soit comme une très petite suite de tests, soit comme un sous-ensemble d'une suite de tests de conformité (incluant des tests de capacité et de comportement). Ils peuvent être utilisés de façon indépendante ou associés à une suite de tests de conformité. L'existence et l'exécution des tests d'interconnexion de base sont optionnelles.

6.1.3 *Tests de capacités*

6.1.3.1 Les tests de capacités permettent de tester de façon limitée la conformité à chacune des conditions de conformité statique d'une Recommandation*, pour déterminer quelles capacités de l'IUT peuvent être observées et pour vérifier que les capacités observables sont valides au regard des conditions de conformité statique, et de la PICS.

6.1.3.2 Les tests de capacités sont appropriés:

- a) pour vérifier aussi loin que possible la cohérence de la PICS avec l'IUT;
- b) comme filtre préliminaire avant d'entreprendre des tests plus approfondis et plus coûteux;
- c) pour vérifier que les capacités de l'IUT répondent aux conditions de conformité dynamique;
- d) pour permettre une sélection efficace des tests de comportement à effectuer pour une IUT particulière;
- e) pris avec des tests de comportement, comme base de déclaration de conformité.

6.1.3.3 Les tests de capacités ne sont pas appropriés:

- a) considérés isolément, comme base de déclarations de conformité par le fournisseur d'une réalisation;
- b) pour des tests détaillés du comportement associé à chaque capacité réalisée ou non réalisée;
- c) pour la solution de problèmes constatés lors de l'utilisation réelle ou quand d'autres tests indiquent une éventuelle non-conformité, bien que les tests de capacité aient été satisfaits.

6.1.3.4 Les tests de capacité sont normalisés en une suite de tests de conformité. Ils peuvent constituer un ou plusieurs groupes de tests indépendants distincts ou ne pas être distingués des tests de comportement.

6.1.4 *Tests de comportement*

6.1.4.1 Les tests de comportement testent une réalisation de façon aussi complète que possible, sur la gamme complète des conditions de conformité dynamique spécifiées dans une Recommandation*. Comme le nombre de combinaisons possibles d'événements et de chronologies d'événements est infini, ces tests ne peuvent pas être exhaustifs. Une autre limitation est que ces tests sont conçus pour être exécutés collectivement dans un même environnement de test, de sorte que tous les échecs difficiles ou impossibles à détecter dans cet environnement risquent d'être omis. Il est donc possible qu'une réalisation non conforme subisse avec succès la suite des tests de conformité; un objectif de la conception d'une suite de tests est de réduire au minimum le nombre d'occasions où ceci risque de se produire.

6.1.4.2 Associés à des tests de capacité, les tests de comportement sont appropriés comme base d'évaluation de conformité.

6.1.4.3 Les tests de comportement ne sont pas appropriés pour la résolution de problèmes constatés lors de l'utilisation réelle ou quand d'autres tests indiquent une éventuelle non-conformité, bien que les tests de comportement aient été satisfaits.

6.1.4.4 Les tests de comportement sont normalisés comme corps d'une suite de tests de conformité.

Remarque – Les tests de comportement comprennent des tests portant sur la validité du comportement de l'IUT en réponse à une mise en œuvre valide, inopportune et syntaxiquement non valide du protocole par le testeur inférieur. Ils comprennent le test du rejet opposé par l'IUT aux tentatives d'utilisation des caractéristiques (capacités) déclarées dans la PICS comme non réalisées. Les tests de capacité n'ont donc pas besoin de comprendre des tests portant sur des capacités omises dans la PICS.

6.1.5 *Tests de résolution de conformité*

6.1.5.1 Les tests de résolution de conformité fournissent des réponses permettant de déterminer, de façon aussi définitive que possible, si une réalisation satisfait des obligations bien précises. Du fait des problèmes d'exhaustivité signalés au § 6.1.4.1, les réponses définitives sont obtenues au prix d'un confinement des tests dans un domaine étroit.

6.1.5.2 L'architecture et la méthode de test seront normalement choisies spécifiquement pour les conditions à tester, elles n'auront pas besoin de présenter une possibilité d'utilisation générale pour d'autres obligations. Elles peuvent même être de celles qui sont considérées comme inacceptables (pour des suites de tests de conformité normalisés): par exemple

comprenant des méthodes spécifiques à une réalisation, utilisant les utilitaires de diagnostic et de mise au point d'un système d'exploitation spécifique.

6.1.5.3 La distinction entre les tests de comportement et les tests de résolution de conformité peut être illustrée par le cas d'un événement tel qu'une réinitialisation. Les tests de comportement peuvent n'inclure qu'une sélection représentative des conditions dans lesquelles une réinitialisation peut se produire, et risquent de ne pas réussir à détecter un comportement incorrect dans d'autres circonstances. Les tests de résolution de conformité seront concentrés sur les conditions dans lesquelles un comportement incorrect a déjà été présumé, et devront confirmer si les présomptions sont ou non fondées.

6.1.5.4 Les tests de résolution de conformité sont appropriés:

- a) pour fournir une réponse par oui ou non, dans une situation strictement délimitée et identifiée antérieurement (par exemple au cours de l'élaboration de la réalisation, pour vérifier si une caractéristique particulière a été correctement réalisée, ou durant l'utilisation opérationnelle, pour rechercher la cause de problèmes);
- b) comme moyen d'identification et comme solution aux insuffisances d'une suite de tests de conformité existante.

6.1.5.5 Les tests de résolution de conformité ne sont pas appropriés comme base de décision de la conformité d'ensemble d'une réalisation.

6.1.5.6 Les tests de résolution de conformité ne sont pas normalisés.

Remarque sur le § 6.1 – Un sous-produit des tests de conformité peut être la détermination d'erreurs et d'insuffisances dans des Recommandations* de protocole.

6.2 *Informations complémentaires sur la mise en œuvre du protocole destinées aux tests (PIXIT)*

Pour tester une mise en œuvre de protocole, le laboratoire de test demandera des informations sur l'IUT et son environnement de test, en plus de celles fournies par la PICS. Ces «informations complémentaires sur la mise en œuvre du protocole destinées aux tests» (PIXIT) seront fournies par le client qui soumet la réalisation aux tests, en concertation avec le laboratoire de test.

Le PIXIT peut contenir les informations suivantes:

- a) les informations nécessaires au laboratoire de test pour pouvoir exécuter la suite de tests appropriée sur le système spécifique (par exemple, des informations concernant la méthode de test à appliquer pour l'exécution des tests élémentaires, des informations d'adressage);
- b) des informations déjà mentionnées dans la PICS et qui doivent être précisées (par exemple, la plage des valeurs d'un délai de temporisation, déclarée comme un paramètre dans la PICS, pourra être spécifiée dans le PIXIT);
- c) des informations permettant de déterminer quelles capacités déclarées prises en charge dans la PICS, sont testables ou non testables;
- d) d'autres informations de nature administrative (par exemple identificateur de l'IUT, la référence à la PICS concernée).

Le PIXIT ne doit pas être en contradiction avec la PICS appropriée.

Le concepteur de la suite de tests abstraite, le réalisateur de l'équipement de tests et le laboratoire de test contribueront tous à l'élaboration du formulaire PIXIT.

6.3 *Schéma général de l'évaluation de conformité*

6.3.1 La principale caractéristique de l'évaluation de conformité est une configuration d'équipement permettant des échanges d'informations entre l'IUT et un testeur réel. Ces échanges sont contrôlés et observés par le testeur réel.

6.3.2 Dans le schéma général conceptuel, le test de conformité doit se faire par étapes, comprenant des revues de conformité statique et des phases de test en fonctionnement réel, l'aboutissement étant l'établissement d'un rapport de test aussi complet que possible.

6.3.3 Ces étapes sont:

- a) l'analyse de la PICS;
- b) la sélection et le paramétrage des tests;
- c) les tests d'interconnexion de base (optionnels);
- d) les tests de capacités;
- e) les tests de comportement;
- f) la revue et l'analyse des résultats des tests;
- g) la synthèse, les conclusions et l'établissement d'un rapport de test de conformité.

Elles sont représentées sur la figure 1/X.290, partie 1.

Avant l'exécution de tout test, la sélection et le paramétrage des tests élémentaires sont opérés d'après la PICS et le PIXIT de l'IUT.

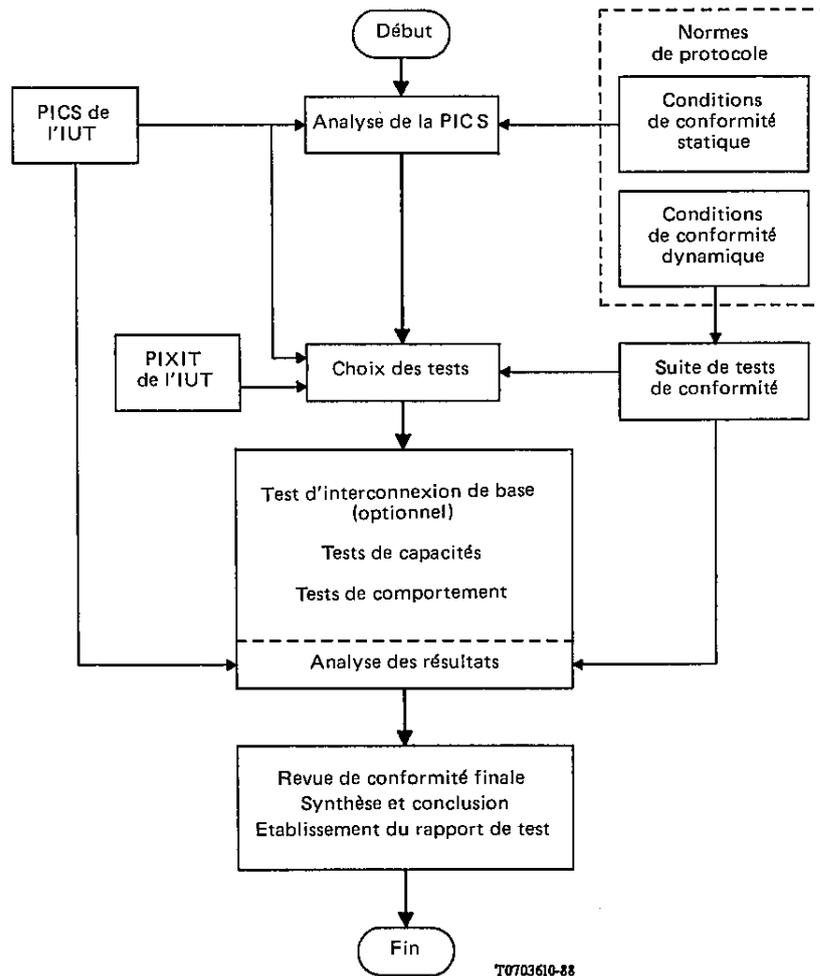


FIGURE 1/X.290, Partie 1

Schéma général de l'évaluation de conformité

6.4 Analyse des résultats

6.4.1 Généralités

6.4.1.1 Résultats et verdicts

La série des événements qui se sont produits durant l'exécution d'un test élémentaire constitue les résultats observés (de l'exécution de ce test); ils comprennent toutes les entrées et sorties de l'IUT aux points de contrôle et d'observation.

Les résultats prévus sont identifiés et définis par la spécification du test élémentaire abstrait et la Recommandation* de protocole. Chaque test élémentaire peut comporter un ou plusieurs résultats prévus, les résultats prévus sont essentiellement définis en termes abstraits.

Un verdict est une déclaration «succès», «échec» ou «non concluant» à associer à chaque résultat prévu de la spécification de la suite de tests abstraite.

L'analyse des résultats est effectuée en comparant les résultats observés avec les résultats prévus.

Le verdict rendu sur un résultat observé est celui associé au résultat prévu correspondant. Si le résultat observé est imprévu, la spécification de la suite de tests abstraite établira alors quel verdict par défaut doit être rendu.

Les moyens avec lesquels la comparaison des résultats observés avec les résultats prévus est effectuée n'entrent pas dans le cadre de la présente Recommandation.

Remarque – Les possibilités sont entre autres:

- a) une comparaison manuelle ou automatisée (ou une combinaison des deux);
- b) une comparaison au moment de l'exécution ou après celle-ci;
- c) la traduction des résultats observés en termes abstraits, à des fins de comparaison avec les résultats prévus, ou la traduction des résultats prévus dans les termes utilisés pour enregistrer les résultats observés.

Le verdict sera succès, échec ou non concluant:

- a) succès signifie que les résultats observés satisfont l'objet du test et sont corrects au regard de la ou des Recommandations* concernées et de la PICS;
- b) échec signifie que le résultat observé est syntaxiquement non valide ou inopportun au regard de la ou des Recommandations* concernées ou de la PICS;
- c) non concluant signifie que les résultats observés sont corrects au regard de la ou des Recommandations* concernées, mais empêchent de remplir l'objet du test.

Le verdict rendu pour un résultat particulier dépendra de l'objet du test et de la validité du comportement observé du protocole.

Les verdicts rendus pour les tests élémentaires individuels seront synthétisés en un résumé d'ensemble pour l'IUT, fondé sur les tests élémentaires exécutés.

6.4.1.2 *Rapports de test de conformité*

Les résultats des tests de conformité seront présentés sous la forme d'un ensemble de rapports de test de conformité. Ces rapports seront de deux types: un rapport de test de conformité du système (SCTR) et un rapport de test de conformité au protocole (PCTR).

Le SCTR, qui sera toujours fourni, donne un résumé d'ensemble de l'état de conformité du SUT, en ce qui concerne son IUT monocouche ou multicouche. La normalisation éventuelle d'un formulaire SCTR nécessite un complément d'étude.

Le PCTR, dont un exemplaire est établi pour chacun des protocoles, pour lequel est testé le SUT, présente l'ensemble des résultats des tests élémentaires, avec des références aux journaux de conformité contenant les résultats observés. Le PCTR donne les références à tous les documents nécessaires concernant la conduite de l'évaluation de conformité pour ce protocole.

Un formulaire PCTR normalisé nécessite un complément d'étude. La liste ordonnée des tests élémentaires à mentionner dans le PCTR sera spécifiée dans la Recommandation* concernant la suite de tests de conformité.

6.4.2 *Reproductibilité des résultats*

Pour atteindre l'objectif de crédibilité des tests de conformité, il est évident que le résultat de l'exécution d'un test élémentaire sur une IUT doit être le même chaque fois qu'il est exécuté. Statistiquement, il peut ne pas être possible d'exécuter une suite complète de tests de conformité et d'observer des résultats parfaitement identiques à ceux obtenus en une autre occasion: des événements imprévus se produisent, et ceci est une caractéristique des environnements concernés. Néanmoins, au niveau du test élémentaire, il est très important que les concepteurs de tests et les laboratoires de test s'efforcent de réduire au minimum le risque qu'un test élémentaire produise des résultats différents en des occasions différentes.

6.4.3 Comparabilité des résultats

Pour atteindre l'objectif ultime des tests de conformité, le résumé d'ensemble concernant la conformité d'une IUT doit être indépendant de l'environnement où les tests ont été exécutés. Autrement dit, la normalisation de toutes les procédures concernées par les tests de conformité devrait se traduire par la possibilité de décerner pour l'IUT un résumé d'ensemble comparable, que le test soit effectué par le fournisseur, un utilisateur ou un laboratoire de test indépendant. Pour réaliser cet objectif, un grand nombre de facteurs sont à étudier, dont certains des plus importants:

- a) une conception précise de la spécification des tests élémentaires abstraits, autorisant une certaine flexibilité quand cela est approprié, mais précisant les conditions qui doivent être remplies (ce qui est l'objet de la présente Recommandation);
- b) une spécification précise du testeur réel qui sera utilisé pour exécuter la suite de tests; là encore, cette spécification doit présenter une certaine flexibilité quand cela est approprié, mais préciser quelles conditions doivent être remplies, en incluant (le cas échéant) toutes les procédures de coordination de tests;
- c) une spécification précise de la procédure à suivre dans la détermination de la façon d'utiliser le contenu de la PICS pour l'analyse des résultats des tests élémentaires; aucune place ne doit être laissée pour une interprétation «optimiste»;
- d) une spécification précise des procédures à suivre par les laboratoires de test en ce qui concerne la répétition d'un test élémentaire avant la formulation d'un verdict final concernant l'objet de ce test;
- e) un formulaire de rapport de test de conformité;
- f) une spécification précise des procédures à appliquer pour la synthèse du résumé d'ensemble.

6.4.4 Possibilité d'audit des résultats

Pour des raisons légales, entre autres, il peut être nécessaire de revoir les résultats observés lors de l'exécution d'une suite de tests de conformité, pour s'assurer que toutes les procédures ont été suivies correctement. Que l'analyse ait été effectuée en mode manuel ou automatique, il est essentiel que toutes les entrées, toutes les sorties et tous les autres événements de test soient soigneusement enregistrés dans un journal, et que l'analyse des résultats soit également enregistrée. Dans certains cas, la responsabilité peut en incomber au réalisateur de l'équipement de test, qui peut choisir d'incorporer des critères de test dans le journal de conformité, ainsi que dans tous les résultats. Dans d'autres cas, elle peut incomber au laboratoire de test, à qui il peut être imposé de suivre toutes procédures standards s'appliquant à l'enregistrement des résultats.

Remarque – En matière d'audit, on pourrait préférer certaines procédures automatiques, mais pour qu'elles puissent être appréciées du point de vue légal, ces procédures automatiques devraient elles-mêmes être homologuées pour être crédibles.

7 Méthodes de test

7.1 Introduction

Le test de la mise en œuvre d'un protocole OSI* donné peut impliquer l'utilisation de plusieurs méthodes de test, car les systèmes à tester peuvent se présenter dans plusieurs configurations, et diffèrent en ce qui concerne leur capacité à donner accès à des effets se produisant à une frontière de couches.

Ce chapitre précise d'abord les caractéristiques du système à tester qui doivent être prises en considération; les méthodes de test possibles sont ensuite définies en termes abstraits; pour conclure, des directives sont données en ce qui concerne leur applicabilité à des systèmes réels.

7.2 Classification des systèmes ouverts réels et des IUT en vue des tests de conformité

7.2.1 Classification des systèmes à tester

7.2.1.1 Il y a une relation entre les méthodes de test et les configurations des systèmes ouverts réels à tester. Les méthodes de test appropriées varient selon:

- a) la principale fonction du système (système d'extrémité ou système relais);
- b) les couches qu'utilisent les protocoles OSI*;
- c) l'autorisation éventuelle de mettre également en œuvre des protocoles non OSI*.

7.2.1.2 Les configurations suivantes de systèmes ont été identifiées à des fins de tests de conformité; elles sont représentées sur les figures 2/X.290, partie 1 à 4/X.290, partie 1. Les configurations 1, 2 et 3 sont les configurations de base de systèmes à tester (SUT):

- a) Configuration 1: système ouvert à sept couches (système d'extrémité)
Ces systèmes mettent en oeuvre des protocoles de Recommandations* OSI* dans les sept couches.
- b) Configuration 2: système ouvert partiel jusqu'au niveau N (système d'extrémité)
Ces systèmes mettent en oeuvre des protocoles de Recommandations* OSI* dans les couches 1 à N.
- c) Configuration 3: systèmes relais ouverts

Ces systèmes mettent en œuvre des protocoles OSI* dans les couches 1 à 3 (systèmes relais de niveau réseau) ou 1 à 7 (systèmes relais de niveau application).

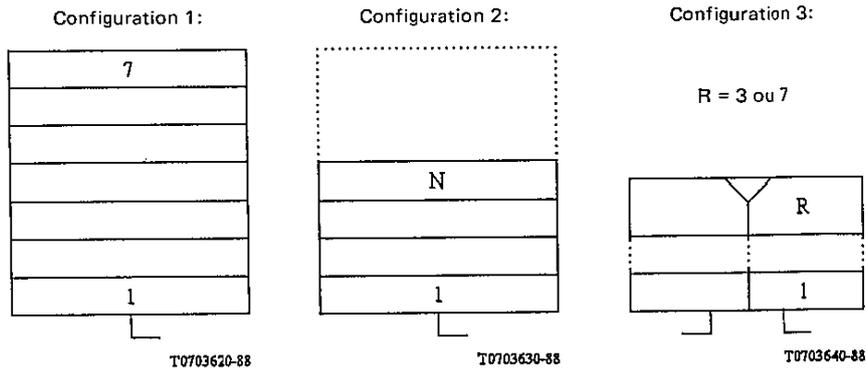


FIGURE 2/X.290, Partie 1
Système ouvert à sept couches

FIGURE 3/X.290, Partie 1
Système ouvert partiel de niveau (N)

FIGURE 4/X.290, Partie 1
Système relais ouvert

7.2.1.3 D'autres configurations peuvent dériver des configurations de base.

Un SUT peut être une combinaison des configurations de base 1 et 2, offrant la possibilité d'utiliser des protocoles OSI* et non OSI* au-dessus de la couche N (voir la figure 5/X.290, partie 1).

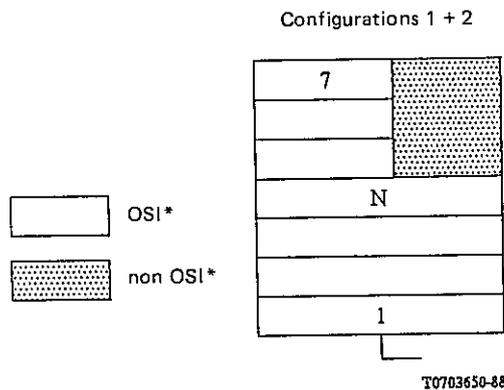


FIGURE 5/X.290, Partie 1
Combinaison de systèmes ouverts à sept couches et partiel de niveau (N)

7.2.2 Identification de la réalisation à tester (IUT)

Une réalisation à tester (IUT) est la partie d'un système ouvert réel qui doit être soumise aux tests de conformité. Elle doit être une mise en œuvre de un ou plusieurs protocoles OSI* adjacents.

Les IUT peuvent être définies pour les configurations 1 et 2 des SUT, comme des IUT monocouches (une seule couche du SUT est à tester), ou comme des IUT multicouches (un ensemble formé d'un nombre quelconque de couches adjacentes du SUT doit être testé conjointement).

Une IUT définie dans un système relais ouvert inclura au moins la couche qui assure la fonction de relais.

Quand des protocoles OSI* et non OSI* coexistent dans un système, la ou les IUT seront définies pour le ou les modes OSI* de fonctionnement. Les tests des protocoles non OSI* n'entrent pas dans le cadre de la présente Recommandation.

Les clients et les laboratoires de test se mettront d'accord sur la partie du SUT à considérer comme IUT.

7.3 *Méthodologie de test abstraite – Considérations générales*

Les méthodes de test doivent se référer à une méthodologie de test abstraite, fondée sur le modèle de référence OSI. Considérons d'abord des systèmes d'extrémité [systèmes ouverts à sept couches ou partiel de niveau (N)] et des IUT monocouches de ces systèmes: des méthodes de test abstraites sont décrites en précisant quels événements sortants de l'IUT peuvent être observés et quels événements entrants peuvent être contrôlés. Plus spécifiquement, une méthode de test abstraite est décrite en identifiant les points les plus voisins de l'IUT, auxquels le contrôle et l'observation doivent s'exercer.

Les Recommandations* de protocole OSI* définissent le comportement autorisé (c'est-à-dire les conditions de conformité dynamique) d'une entité de protocole en termes d'unités de données de protocole (PDU) et de primitives de service abstraites (ASP) échangées au-dessus et en dessous de cette entité. Le comportement d'une entité (N) est ainsi défini en fonction d'ASP(N) et d'ASP(N – 1), ces dernières comprenant les PDU (N).

Si une IUT comprend plusieurs entités de protocole, le comportement requis peut être défini en termes d'ASP échangées au-dessus et en dessous de cette IUT en y incluant les PDU de ces protocoles.

Le point de départ de l'élaboration de méthodes de test est l'architecture de test conceptuelle, présentée dans la figure 6/X.290, partie 1. Cette architecture est une «boîte noire» active de test, fondée sur la définition du comportement requis de l'IUT.

L'action du testeur représentée dans la figure 6/X.290, partie 1 peut être exercée localement, auquel cas il existe un couplage direct à l'intérieur du système à tester, ou extérieurement via une liaison ou un réseau de communication. Les deux ensembles d'interactions, exercés au dessus et en dessous de l'unité, peuvent en pratique être observés et contrôlés à partir de plusieurs points différents, localement ou extérieurement.

Les points de contrôle et d'observation (PCO possibles) sont déterminés par trois facteurs:

- a) l'observation et le contrôle s'exercent-ils sur les ASP ou sur les PDU?
- b) le niveau de la couche des ASP ou des PDU concernées;
- c) ces ASP ou PDU sont-elles contrôlées et observées à l'intérieur du système à tester ou dans un système distant du système à tester? Dans le second cas, les ASP sont distinguées par l'ajout d'un caractère guillemets doubles (").

Les PCO possibles à l'intérieur du SUT sont représentés sur la figure 7a)/X.290, partie 1. Les PCO possibles, quand l'activité exercée en dessous de l'IUT est contrôlée et observée extérieurement, sont représentés sur la figure 7b)/X.290, partie 1. Il peut être constaté sur ces figures qu'une multiplicité de PCO est possible dans différentes couches, offrant différents niveaux de contrôle et d'observation du comportement d'une IUT. La présente Recommandation opère une sélection à partir de cet ensemble de PCO possibles, pour définir un nombre limité de méthodes de test abstraites.

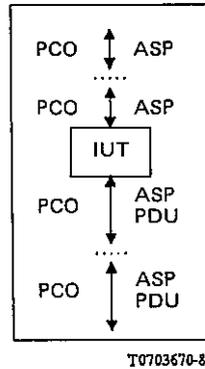


FIGURE 7 a)/X.290, Partie 1
PCO possibles dans un SUT

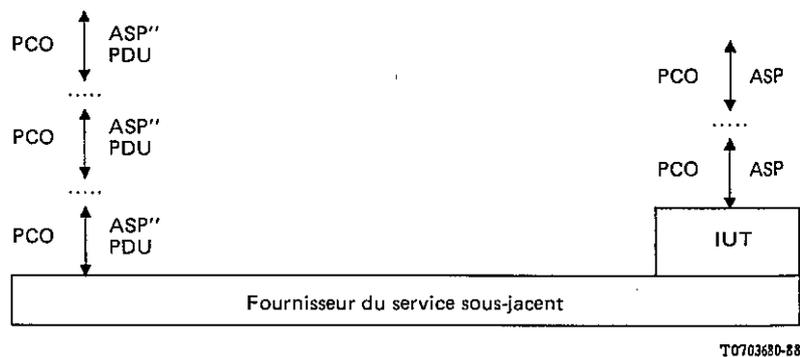


FIGURE 7 b)/X.290, Partie 1
PCO possibles pour des tests externes

Si le contrôle et l'observation exercés en dessous de l'IUT ou à l'extérieur de celle-ci, sont spécifiés en termes d'ASP, ils devront inclure le contrôle et l'observation des PDU véhiculées par ces ASP; mais, s'ils sont spécifiés en termes de PDU [au niveau de la couche (N)], les ASP (au niveau de la couche N – 1) ne sont pas considérées comme contrôlées ou observées.

Il est supposé que l'activité des ASP exercée en dessous de l'IUT est observable et contrôlable, au moins par l'intermédiaire de l'activité homologue d'un système distant participant au test – c'est-à-dire les ASP «correspondantes». Quand les ASP échangées en dessous de l'IUT ne sont ni contrôlables ni observables localement, les tests de conformité peuvent ainsi être effectués extérieurement, à condition que le service sous-jacent fourni soit suffisamment fiable pour permettre un contrôle et une observation à distance.

Il est possible que l'activité des ASP échangées au-dessus de l'IUT ne soit ni contrôlable ni observable, auquel cas cette activité est dite «cachée».

Il n'est pas imposé au SUT de donner accès aux frontières de ses couches. Toutefois, la possibilité de fournir un tel accès et les positions possibles de ces frontières par rapport aux couches de l'IUT, sont des facteurs à prendre en considération dans la définition des méthodes de test, qui peuvent tirer parti de cet accès pour définir les suites des tests en termes d'ASP correspondantes. Peu importe si l'accès aux frontières accessibles se fait par l'intermédiaire des points d'accès aux services (SAP) ou d'autres PCO.

La figure 8/X.290, partie 1 présente des exemples d'IUT avec diverses possibilités d'accès aux frontières de couches.

Remarque – Une Recommandation* de suite de tests de conformité peut, en outre, définir des «primitives locales abstraites». Elles sont utilisées pour spécifier le contrôle et l'observation d'événements ou d'états auxquels il est fait référence dans la Recommandation* de protocole, mais qui sont internes à l'IUT et qui ne peuvent pas être exprimés en termes d'ASP. Ce sont des abréviations de descriptions textuelles de contrôle et d'observation à exercer par le testeur supérieur.

Des considérations similaires s'appliquent aux systèmes relais (pour de plus amples détails, voir la partie 2 de la présente Recommandation).

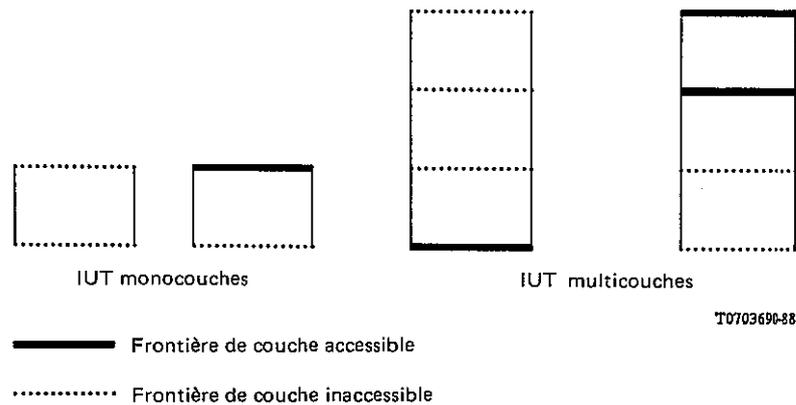


FIGURE 8/X.290, Partie 1
Exemples de configuration d'IUT

7.4 Fonctions de test abstraites

La définition des méthodes de test abstraites nécessite la répartition des PCO entre deux fonctions de test abstraites, le testeur inférieur et le testeur supérieur.

Le testeur inférieur est la représentation abstraite des moyens d'exercer, au cours de l'exécution du test, le contrôle et l'observation au PCO approprié, situé en dessous ou à distance de l'IUT, comme spécifié par la méthode de test abstraite choisie. Il constitue donc la fonction de test associée au contrôle et à l'observation de la frontière inférieure de l'IUT. Si l'action du testeur est locale au SUT, le testeur inférieur prendra la place de la partie inférieure du SUT. Si l'action du testeur inférieur est extérieure au SUT, il se fondera sur le service (N – 1), fourni conjointement par le testeur inférieur lui-même, une liaison de communication et le SUT.

Le testeur supérieur est une représentation abstraite des moyens de fournir, durant l'exécution du test, le contrôle et l'observation de la frontière de service supérieure de l'IUT et de toute primitive locale abstraite concernée.

Une coopération entre le testeur supérieur et le testeur inférieur est nécessaire; les règles de coopération sont appelées les procédures de coordination de tests.

Les méthodes de test varient selon la façon dont elles spécifient les procédures de coordination de tests. Dans certains cas, il est possible de définir un protocole de gestion de tests assurant la coordination entre les testeurs supérieur et inférieur. Dans d'autres cas, il est seulement possible de décrire les conditions à remplir par les procédures de coordination de tests, sans spécifier quels mécanismes doivent être utilisés pour les réaliser.

7.5 Présentation générale des méthodes de test abstraites

7.5.1 IUT de systèmes d'extrémité

Pour les IUT définies à l'intérieur de SUT systèmes d'extrémité (configurations 1 et 2 des figures 2/X.290, partie 1 et 3/X.290, partie 1), quatre catégories de méthode de test abstraite sont définies, une locale, et trois externes qui supposent que le testeur inférieur est situé à distance du SUT et lui est connecté par une liaison de communication ou un réseau.

7.5.2 Les méthodes de test locales

Dans les méthodes de test abstraites locales, les PCO sont par définition aux frontières de services, situés au-dessus et en dessous de l'IUT. Les événements de test sont spécifiés en termes d'ASP échangées au-dessus de l'IUT et d'ASP et de PDU échangées en dessous de l'IUT, comme représenté sur la figure 9a)/X.290, partie 1. D'un point de vue abstrait, un testeur inférieur est considéré observer et contrôler les ASP et les PDU échangées en dessous de l'IUT alors qu'un testeur supérieur observe et contrôle les ASP échangées au-dessus de l'IUT. Les conditions à remplir par les procédures de coordination de tests utilisées pour coordonner les réalisations des testeurs supérieur et inférieur sont définies dans les suites de tests de conformité abstraites, mais pas les procédures de coordination de tests.

7.5.3 Méthodes de test externes

Les méthodes de test externes utilisent le contrôle et l'observation des ASP échangées en dessous de l'IUT, au moyen d'un testeur inférieur séparé du SUT, ainsi que le contrôle et l'observation des ASP échangées au-dessus de l'IUT. Trois catégories de méthodes de test abstraites externes sont définies: les méthodes de test réparties, coordonnées et à distance. Elles diffèrent selon le niveau d'exigence ou de normalisation des procédures de coordination de tests, l'accès à la frontière de couche située au-dessus de l'IUT et les conditions imposées au testeur supérieur. Elles sont représentées sur les figures 9b), 9c) et 9d)/X.290, partie 1.

La méthode de test coordonnée exige que les procédures de coordination de tests utilisées pour coordonner la réalisation des testeurs supérieur et inférieur soient réalisées au moyen de protocoles de gestion de tests. Les deux autres méthodes ne font aucune supposition quant à la réalisation des procédures de coordination de tests.

Les méthodes de test répartie et coordonnée nécessitent des fonctions spécifiques de la part du testeur supérieur, situé au-dessus de l'IUT. Ce n'est pas le cas de la méthode à distance.

La méthode répartie nécessite l'accès à la frontière supérieure de l'IUT, mais pas les deux autres méthodes.

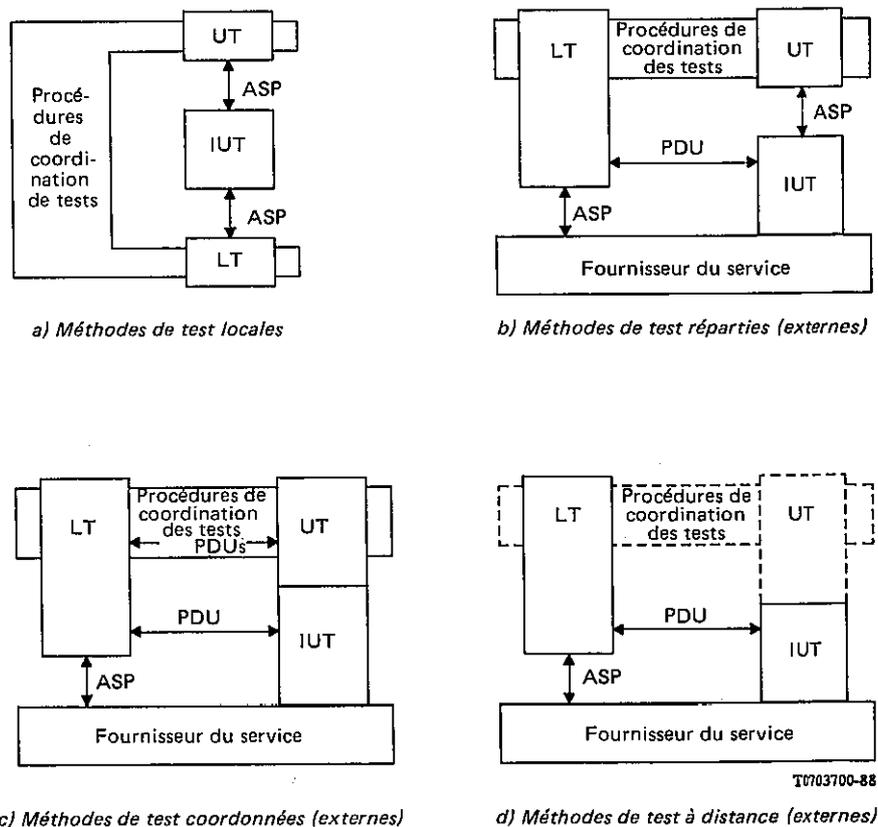


FIGURE 9/X.290, Partie 1

Présentation générale des méthodes de test abstraites

7.5.4 *Variantes des méthodes de test de systèmes d'extrémité*

Chaque catégorie de méthodes de test comporte des variantes qui peuvent être appliquées à des IUT monocouches ou à des IUT multicouches. Des variantes imbriquées peuvent être utilisées pour les IUT multicouches, dans lesquelles la mise en œuvre des protocoles doit être testée couche par couche.

Toutes les méthodes de test abstraites de systèmes d'extrémité sont intégralement spécifiées dans la section 8 de la partie 2 de la présente Recommandation, y compris leurs éventuelles variantes monocouches, multicouches et encastrées.

7.5.5 *IUT de systèmes relais*

Deux méthodes de test sont définies pour les systèmes relais ouverts: les méthodes en boucle et les méthodes transversales. Elles sont intégralement spécifiées dans la section 8 de la partie 2 de la présente Recommandation.

7.6 *Applicabilité des méthodes de test à des systèmes ouverts réels*

L'architecture et l'état d'élaboration d'un système ouvert réel déterminent les méthodes de test qui lui sont applicables.

Les méthodes de test locales sont utilisées pour les systèmes en cours d'élaboration, quand leur architecture permet d'isoler une IUT, qu'elle soit monocouche ou multicouche.

Les méthodes de test externes sont utilisées pour les tests complets ou partiels de systèmes d'extrémité qui peuvent être connectés à des réseaux de communication.

Les méthodes de test coordonnées s'appliquent quand il est possible de mettre en œuvre un protocole de gestion de tests normalisé dans un testeur supérieur, à l'intérieur du SUT et au-dessus de l'IUT.

Les méthodes de test à distance s'appliquent quand il est possible d'utiliser certaines fonctions du SUT pour contrôler l'IUT durant les tests, au lieu d'utiliser un testeur supérieur.

Les méthodes de test réparties s'appliquent quand il est nécessaire de laisser une liberté complète de réalisation des procédures de coordination des tests, entre le SUT et le testeur inférieur, mais de spécifier en détail les conditions de contrôle et d'observation aux deux extrémités.

Les méthodes de test monocouches sont les plus appropriées pour tester la majorité des conditions de conformité à un protocole.

Les méthodes de test multicouches seront utilisées pour tester de véritables conditions de conformité dynamique multicouches.

Les méthodes de test encastrées permettent l'application de tests monocouches à toutes les couches d'une IUT multicouche.

Pour les systèmes ouverts à sept couches, on appliquera de préférence et de façon incrémentielle, les méthodes encastrées monocouches externes appropriées, avec les PCO suivants:

- a) l'interface supérieure de la couche application, telle qu'elle est présentée par le système ouvert à sept couches, quand cela est applicable;
- b) chaque SAP situé au-dessous du protocole sur lequel est axé le test (ou le PCO correspondant s'il n'existe pas de tels SAP), tel que contrôlé et observé à partir du testeur inférieur externe, en commençant par le protocole de plus bas niveau de l'IUT et en passant aux niveaux supérieurs successifs.

7.7 *Applicabilité des méthodes de test aux protocoles et couches OSI**

Les méthodes de test définies dans la présente Recommandation s'appliquent à toutes les couches, sauf à la couche physique et aux protocoles de contrôle d'accès au support, qui n'entrent pas dans le cadre de la présente Recommandation. L'appendice I de la présente Recommandation donne des directives concernant l'applicabilité des méthodes de test à toutes les autres couches.

8 Suites de tests

8.1 *Structure*

Les suites de tests ont une structure hiérarchisée (voir la figure 10/X.290, partie 1), dont un niveau important est le test élémentaire. Chaque test élémentaire a un objet défini avec précision, comme la vérification que l'IUT a telle

capacité requise (par exemple l'aptitude à prendre en charge certaines tailles de paquets), ou présente tel comportement requis (par exemple se comporte comme spécifié quand un événement particulier se produit dans un certain état).

Dans une suite de tests, l'imbrication de groupes de tests permet d'ordonner de façon logique tous les tests élémentaires. Les groupes de tests peuvent être imbriqués jusqu'à n'importe quelle profondeur. Ils peuvent être utilisés pour faciliter la planification, l'élaboration, la compréhension et l'exécution de la suite de tests.

Les tests élémentaires sont structurés en subdivisions appelées modules de tests. Chaque test élémentaire comprend au moins un module de tests: la séquence des événements impliqués par l'objet du test (le «corps du test»). Il peut inclure d'autres modules de tests destinés à mettre l'IUT dans l'état nécessaire pour commencer le corps du test («le préambule») ou dans l'état de repos où elle doit revenir quand le corps du test est terminé («l'épilogue»).

Pour des raisons pratiques, des modules de tests communs peuvent être regroupés en des bibliothèques de modules de tests. Ces bibliothèques peuvent être structurées en ensembles imbriqués de modules de tests, jusqu'à n'importe quelle profondeur d'imbrication. Les bibliothèques de modules de tests peuvent être associées à la suite de tests complète, à un groupe de tests particulier ou à un test élémentaire.

En outre, tous les modules de tests consistent en une séquence d'autres modules de tests et/ou d'événements de test (par exemple le transfert d'une seule PDU ou ASP vers l'IUT ou en provenance de celle-ci). Tous les modules de tests sont donc équivalents à une séquence d'événements de tests (après expansion des modules de tests intérieurs).

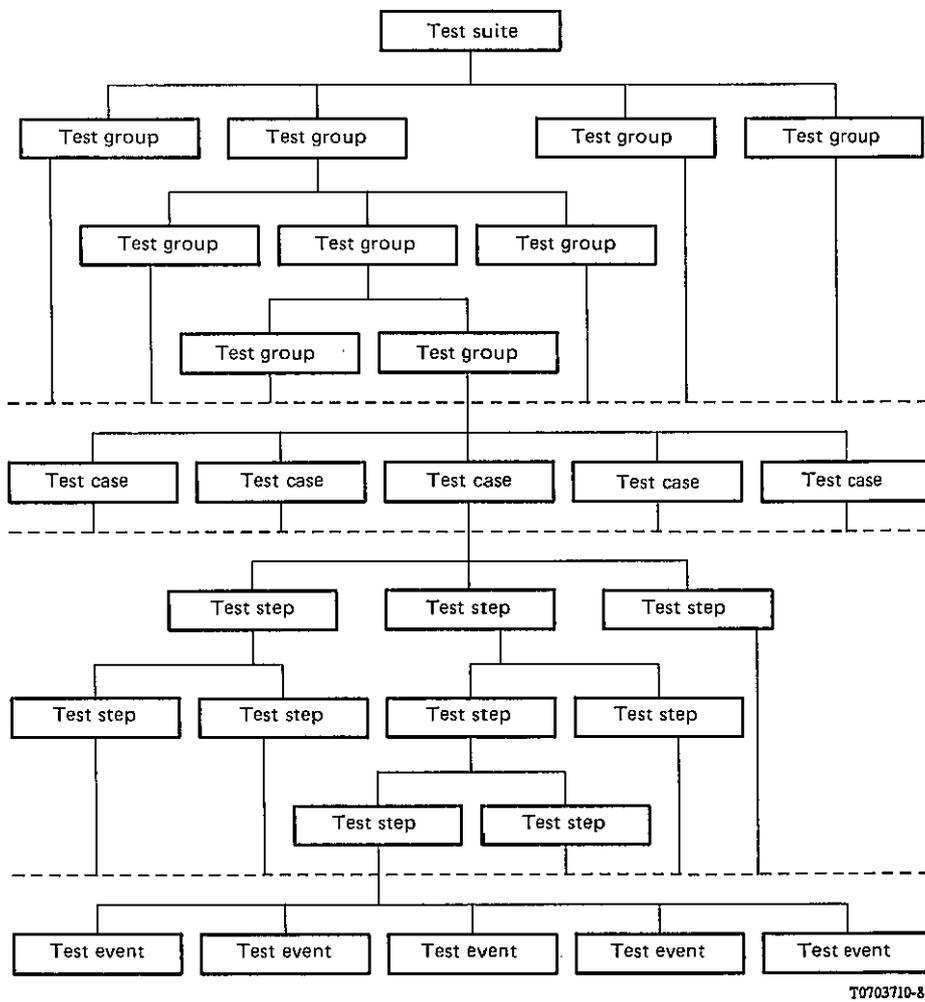


FIGURE 10/X.290, Part 1
Test suite structure

8.2 *Tests élémentaires génériques, abstraits et exécutables*

8.2.1 Un test élémentaire générique est un test qui:

- a) fournit un affinage de l'objet du test;
- b) spécifie l'intégralité des séquences d'événements de test (chemins) du corps du test, qui correspondent à des verdicts «succès», «échec» et «non concluant», en utilisant une notation spéciale;
- c) est utilisé comme racine commune des tests élémentaires abstraits correspondants de différentes suites de tests abstraites du même protocole;
- d) inclut, en guise de préambule, une description de l'état initial à partir duquel le corps du test devra démarrer;
- e) ne nécessite pas de description de l'épilogue;
- f) est spécifié conformément au style des méthodes de test monocouches distante ou répartie.

8.2.2 Un test élémentaire abstrait dérive d'un test générique et de la spécification de protocole concernée; il:

- a) spécifie le test élémentaire dans les termes d'une méthode de test particulière;
- b) ajoute une spécification plus précise des séquences d'événements qui sont décrites de façon seulement informelle dans le test élémentaire générique;
- c) ajoute les séquences d'événements de test nécessaires à remplir les objets du préambule et de l'épilogue du test élémentaire générique, en utilisant une notation spéciale.

8.2.3 Un test élémentaire exécutable dérive d'un test élémentaire abstrait et est d'une forme qui permet son exécution sur un testeur réel, pour tester une réalisation réelle.

8.2.4 Les termes générique, abstraite et exécutable sont utilisés pour décrire les suites de tests qui comprennent respectivement des tests élémentaires génériques, abstraits et exécutables.

8.2.5 Une suite de tests générique couvre l'ensemble ou un sur-ensemble de toutes les suites de tests abstraites possibles pour un protocole particulier.

9 **Relations entre concepts et rôles**

La figure 11/X.290, partie 1 est une représentation schématique des relations entre les diverses Recommandations, l'élaboration de suites de tests génériques, abstraites et exécutables et l'établissement de rapports de test.

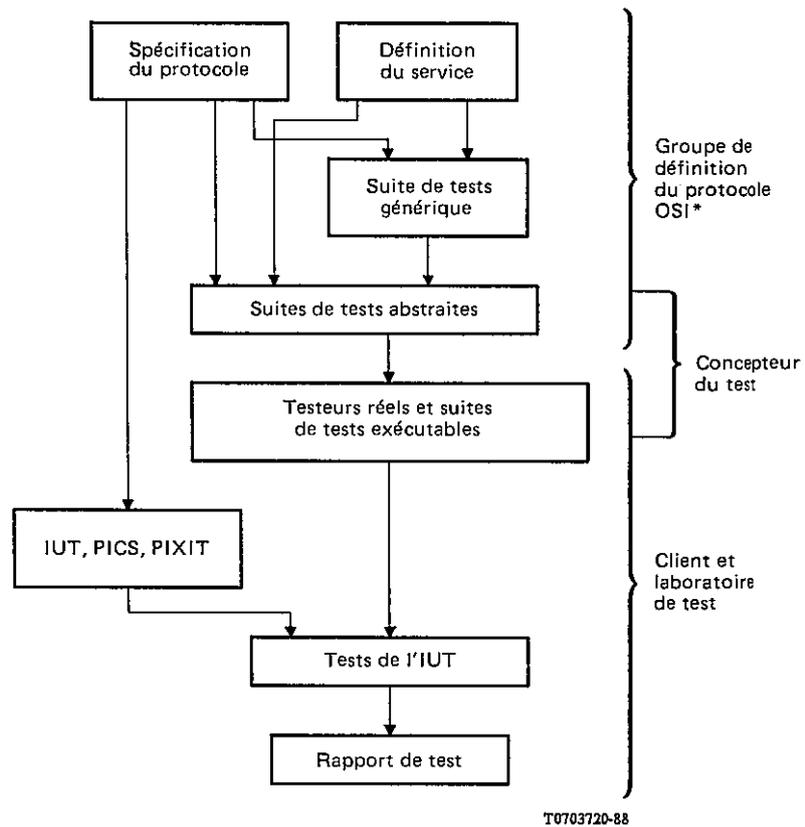
La partie 2 concerne la production des Recommandations relatives aux protocoles testables et des Recommandations relatives aux suites de tests abstraites. La partie 1 fournit des concepts généraux et des définitions.

Remarque – Les autres aspects de l'évaluation de conformité, tels que la dérivation de tests exécutables, la préparation de l'IUT, de la PICS et des PIXIT par le client et le rôle du laboratoire de test, nécessitent un complément d'étude.

10 **Respect des conditions spécifiées**

Dans cette Recommandation, la notion de «respect» correspond au fait de satisfaire les conditions qui y sont spécifiées. Ce mot est utilisé pour essayer d'éliminer la confusion entre le *respect* de la présente Recommandation, et la *conformité* d'une mise en œuvre de protocole à l'une des Recommandations de protocole.

La présente partie de la Recommandation ne contient pas de conditions de satisfaction.



Méthodologie et cadre des tests de conformité OSI:
Concepts généraux

FIGURE 11/X.290, Partie 1
Relations entre concepts et rôles

Partie 2 – Spécification des suites de tests abstraites

0 Introduction

Cette partie de la présente Recommandation présente une méthode commune de spécification des suites de tests de conformité «OSI ou des séries associées X ou T du CCITT» (désignée ci-après par l'abréviation «OSI*») à un niveau qui est indépendant des moyens d'exécution desdites suites de tests (appelées ci-après «suites de tests abstraites»). Ce niveau d'abstraction est approprié à la normalisation et facilite la comparaison des résultats produits par différents organismes qui exécutent les suites de tests exécutables correspondantes.

Le § 1 rappelle l'existence d'obligations auxquelles les concepteurs de protocoles OSI* doivent se conformer pour rendre possible une base objective d'élaboration de suites de tests abstraites. La nécessité de cohérence entre les sections relatives à la conformité et les formulaires PICS et PIXIT des Recommandations* de protocole OSI* est exprimée.

Le § 2 décrit le processus d'élaboration d'une suite de tests abstraite, y compris les critères de conception à appliquer et des directives concernant la structure et la portée de cette suite. Les méthodes de tests abstraites possibles sont définies et des directives sont formulées à l'intention des concepteurs de suites de tests pour les aider à choisir la ou les méthodes à appliquer pour élaborer une suite de tests particulière. Les relations entre les suites de tests abstraites présentes pour différentes méthodes sont explicitées par une suite de tests génériques, indépendante de la méthode de test. Une notation de test est définie, et des obligations et directives sont formulées concernant son utilisation pour la spécification de tests élémentaires génériques et abstraits, cela incluant la subdivision des tests élémentaires en modules de test et l'énoncé de verdicts sur les résultats.

Le concepteur d'une suite de tests est également tenu de fournir des informations au réalisateur de l'équipement de test, en particulier sur les contraintes régissant la sélection et le séquençement des tests élémentaires.

Enfin, des obligations et des directives sont formulées en ce qui concerne la mise à jour des suites de tests.

1 Objectif et domaine d'application

Cette partie de la présente Recommandation spécifie les obligations et donne des directives concernant l'élaboration de suites de tests de conformité à une ou plusieurs Recommandations* OSI, indépendantes du système*. Elle concerne, en particulier, l'élaboration de toutes les Recommandations de suites de tests de conformité à des protocoles OSI*.

Elle s'applique à l'élaboration de tests élémentaires de conformité qui vérifient l'adéquation d'une réalisation aux conditions de conformité statique ou dynamique pertinentes, en contrôlant et en observant le comportement de la mise en œuvre du protocole. Les méthodes de test abstraites incluses dans la présente Recommandation sont en fait utilisables pour spécifier tout test élémentaire qui peut être exprimé de façon abstraite en termes de contrôle et d'observation d'unités de données de protocole, de primitives de service abstraites et de primitives locales abstraites. Néanmoins, certains protocoles peuvent nécessiter des tests élémentaires qui ne peuvent pas être exprimés en ces termes. La spécification de tels tests élémentaires n'entre pas dans le cadre de la présente Recommandation, mais lesdits tests élémentaires peuvent être inclus dans une Recommandation* de suites de tests de conformité.

Remarque – Par exemple, certaines conditions de conformité statique relatives à un service d'application peuvent nécessiter des techniques de test qui sont spécifiques à l'application concernée.

L'élaboration de suites de tests de conformité à des protocoles multi-homologues ou de la couche physique n'entre pas dans le cadre de la présente Recommandation.

Les relations entre les spécifications de tests abstraits et les techniques de descriptions formelles n'entrent pas dans le cadre de la présente Recommandation.

2 Références

- Recommandation X.200 *Modèle de référence de l'interconnexion des systèmes ouverts pour les applications du CCITT* (voir également ISO 7498).
- Recommandation X.214 *Définition du service de transport pour l'interconnexion des systèmes ouverts (OSI) dans les applications du CCITT* (voir également ISO 8072).
- Recommandation X.224 *Spécification du protocole de transport pour l'interconnexion des systèmes ouverts (OSI) pour les applications du CCITT* (voir également ISO 8073).
- Recommandation X.210 *Conventions relatives à la définition de service des couches de l'interconnexion des systèmes ouverts pour les applications du CCITT* (voir également ISO TR 8509).

- Recommandation X.208 *Spécification de la syntaxe abstraite numéro un (ASN.1)* (voir également ISO 8824).
- Recommandation X.209 *Spécification des règles de codage pour la notation de syntaxe abstraite numéro un (ASN.1)* (voir également ISO 8825).
- Recommandation X.290/1 *Méthodologie et cadre des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications du CCITT – partie 1: concepts généraux.*
- Norme ISO 8574-4 *Systèmes de traitement de l'information – Interconnexion des systèmes ouverts – Transfert, accès et gestion de fichiers – partie 4: spécification du protocole de fichiers.*

3 Définitions

Toutes les définitions de la partie 1 de la présente Recommandation s'appliquent aux fins de la présente partie de la Recommandation.

4 Abréviations

Les abréviations données au § 4 de la partie 1 s'appliquent. Les abréviations suivantes s'appliquent également à la présente Recommandation.

ASP*	la primitive de service abstraite, échangée du côté du fournisseur du service distant de l'unité
CM	multicouche coordonnée (méthode de test)
CS	monocouche coordonnée (méthode de test)
CSE	monocouche encastrée coordonnée (méthode de test)
DM	multicouche répartie (méthode de test)
DS	monocouche répartie (méthode de test)
DES	monocouche encastrée répartie (méthode de test)
FDT	technique de description formelle
LM	multicouche locale (méthode de test)
LS	monocouche locale (méthode de test)
LSE	monocouche encastrée locale (méthode de test)
RM	multicouche à distance (méthode de test)
RS	monocouche à distance (méthode de test)
RSE	monocouche encastrée à distance (méthode de test)
TTCN	notation combinée arborescente et tabulaire
YL	en boucle (méthode de test)
YT	transversale (méthode de test)

5 Respect des conditions spécifiées

5.1 Une Recommandation* de protocole qui satisfait à cette partie de la présente Recommandation satisfera toutes les conditions stipulées dans la section 1.

Remarque – Une telle satisfaction est une condition préalable à l'utilisation de la Recommandation* de protocole comme base effective de tests de conformité de réalisations.

5.2 Une suite de tests abstraite qui satisfait cette partie de la présente Recommandation:

- est une suite de tests de conformité;
- doit être spécifiée dans une notation de test normalisée par l'ISO ou le CCITT;
- satisfera toutes les conditions stipulées dans la section 2.

5.3 Il est recommandé d'utiliser comme notation de test la notation combinée arborescente et tabulaire (TTCN). Si TTCN est utilisée, la suite de tests abstraite satisfait toutes les conditions de l'annexe D.

6 Conditions de conformité des Recommandations* OSI*

6.1 Introduction

La signification de la conformité dans l'OSI est traitée dans la partie 1 de la présente Recommandation. Une compréhension non ambiguë et objective des conditions de conformité d'une Recommandation* de protocole ou de syntaxe de transfert OSI* est nécessaire comme condition préalable à l'élaboration d'une suite de tests abstraite pour cette Recommandation*. La présente section stipule les obligations imposées au concepteur de protocole pour garantir une telle compréhension des conditions de conformité.

6.2 Obligations générales

6.2.1 Une distinction claire doit être faite entre les conditions de conformité statique et dynamique. Pour éviter toute ambiguïté, elles devront être stipulées séparément.

6.2.2 La signification de la conformité à la Recommandation doit être claire: ce qu'il faut faire, ce qui est autorisé mais pas obligatoire, et ce qu'il ne faut pas réaliser, pour être conforme à la Recommandation*.

6.2.3 Il doit toujours être possible de décider si une occurrence de communication est dynamiquement conforme ou non.

Par exemple, on doit être capable d'analyser l'enregistrement de l'activité d'une PDU et de décider si elle est valide.

6.2.4 Les obligations en matière d'établissement d'une PICS et de son contenu doivent être stipulées indépendamment des conditions concernant la mise en œuvre du protocole proprement dit.

6.3 Sections relatives à la conformité

6.3.1 Chaque Recommandation* de protocole et de syntaxe de transfert OSI* doit inclure une section relative à la conformité, qui doit être claire et sans ambiguïté.

6.3.2 Les sections relatives à la conformité doivent distinguer les catégories suivantes d'informations qu'elles peuvent contenir:

- a) références à des sections qui énoncent des conditions de conformité dynamique;
- b) conditions de conformité statique concernant la mise en œuvre du protocole;
- c) conditions de conformité statique concernant les dépendances multicouches;
- d) ce qui doit être déclaré dans la PICS concernant b) ci-dessus;
- e) ce qui doit être déclaré dans la PICS concernant c) ci-dessus;
- f) quelles autres informations doivent être fournies (par exemple, pour aider les tests), et si elles doivent se trouver dans la PICS ou ailleurs.

6.3.3 Dans les Recommandations* de protocole en mode connexion, la section de conformité doit également inclure:

- a) l'option de prise en charge de l'initialisation d'une connexion, de l'acceptation d'une connexion ou de l'une et de l'autre;
- b) l'obligation d'être capable d'accepter toutes les séquences correctes de PDU reçues d'entités homologues et d'y répondre par des séquences correctes de PDU, appropriées à l'état défini de la connexion;
- c) l'obligation d'être capable de répondre correctement à toutes les séquences incorrectes de PDU reçues, la réponse étant appropriée à l'état défini de la connexion.

6.4 Directives complémentaires concernant les Recommandations de nouveaux protocoles*

Bien que les Recommandations* de protocole existantes puissent être améliorées par l'élaboration d'un additif permettant d'ajouter un formulaire PICS et d'aligner la section de conformité sur les obligations énoncées ci-dessus, il ne serait pas réaliste d'en attendre une grande amélioration. De nouvelles Recommandations de protocole devraient donc être élaborées, selon les directives complémentaires de l'annexe B, pour faciliter la compréhension des conditions de conformité et éviter toute ambiguïté.

7 Formulaires PICS

7.1 Obligations concernant les formulaires PICS

7.1.1 Les obligations spécifiques imposées aux fournisseurs en matière de PICS à fournir doivent normalement être stipulées dans la Recommandation de protocole concernée. La spécification de ces obligations doit inclure un formulaire PICS. Dans des circonstances exceptionnelles, le formulaire PICS peut se trouver dans une Recommandation* de suite de tests abstraite, et non dans la Recommandation* de protocole. C'est en particulier le cas quand le formulaire PICS doit porter sur différentes versions du même protocole provenant de l'ISO et du CCITT. Normalement, le formulaire PICS devrait se trouver dans une annexe de la Recommandation* de protocole, la section de conformité y faisant référence, complétée, le cas échéant, par un additif plutôt que par une partie de la Recommandation* initiale.

Remarque – La PICS d'une mise en œuvre spécifique d'un protocole doit être accompagnée des informations administratives et documentaires concernant le fournisseur, le système et son environnement prévu.

7.1.2 Le formulaire PICS doit avoir la forme d'un questionnaire ou d'une liste de contrôle, à remplir par le fournisseur ou le réalisateur d'une mise en œuvre du protocole OSI* concerné.

7.1.3 Le formulaire PICS doit couvrir toutes les fonctions, éléments de procédure, paramètres, options, PDU, temporisateurs, dépendances multicouches et autres capacités identifiées dans la Recommandation* de protocole, y compris les capacités optionnelles et conditionnelles. Il est souhaitable d'inclure, quand cela est praticable, toutes les caractéristiques obligatoires. Il doit exister une correspondance bien définie entre les conditions de conformité statique et le formulaire PICS, qui devront présenter une cohérence réciproque.

7.1.4 Le formulaire PICS doit être précédé d'une section qui déclarera:

«Le fournisseur d'une mise en œuvre du protocole déclarée conforme à la présente Recommandation doit remplir le formulaire de déclaration de conformité à la mise en œuvre du protocole (PICS) suivant et doit fournir les informations nécessaires à identifier de façon univoque le fournisseur et la réalisation.»

7.1.5 Le nom, la version et la date de la Recommandation* de protocole visée doivent être indiqués sur chaque page du formulaire PICS.

7.1.6 Le formulaire PICS d'un protocole spécifique doit contenir:

- a) des explications des symboles spéciaux, abréviations, termes spéciaux, avec les références appropriées;
- b) des instructions explicites concernant la mise à jour et l'interprétation de la PICS;
- c) une clause imposant de mentionner toutes les caractéristiques obligatoires qui n'ont pas été réalisées, avec les justifications appropriées;
- d) un ou plusieurs tableaux (ou autres types de formulaires si nécessaire) à remplir pour déclarer en détail les capacités d'une réalisation, incluant:
 - 1) nom de la fonction, type de PDU, temporisateur, paramètre et autre capacité;
 - 2) une colonne où il est précisé si chacun de ces éléments est obligatoire, optionnel, négociable ou conditionnel;
 - 3) si possible, une colonne donnant les références aux sections concernées de la Recommandation*;
 - 4) une colonne donnant, le cas échéant, la plage de valeurs autorisée;
 - 5) une colonne à remplir pour indiquer, le cas échéant, les valeurs ou la plage de valeurs prises en charge;
 - 6) une colonne à remplir pour déclarer si chaque capacité a été réalisée;
- e) le formulaire doit donner une indication claire des types de données à fournir de préférence en réponse (par exemple, bases de numération, types de chaîne, octets, bits, secondes, minutes, etc.).

7.1.7 Le formulaire PICS doit utiliser, si approprié, les abréviations suivantes, sauf si elles créent des confusions avec des abréviations utilisées dans la Recommandation* de protocole spécifique:

- m obligatoire
- o optionnel
- c conditionnel
- n négociable (à l'aide du protocole)
- x exclusion de capacité
- non applicable
- s habilitation à émettre
- r habilitation à recevoir

7.2 *Directives concernant les formulaires PICS*

L'appendice III donne des exemples assez généraux, en guise de directives pour l'élaboration de formulaires PICS.

8 Processus d'élaboration d'une suite de tests

Pour présenter les obligations et les directives générales concernant la spécification d'une suite de tests abstraite, il est utile de supposer une forme normale du processus de production d'une suite de tests. Ce paragraphe décrit le processus quand il revêt une telle forme normale. Les concepteurs de suites de tests abstraites ne sont pas obligés de suivre exactement cette forme normale, mais il leur est recommandé d'adopter une démarche similaire comprenant les mêmes étapes, éventuellement dans un ordre différent.

Pour les applications de la présente Recommandation, le processus de production d'une suite de tests abstraite est supposé être le suivant:

- a) étude de la ou des Recommandations* concernées pour déterminer quelles sont les conditions de conformité (y compris les options) qui doivent être testées, et celles qui doivent être déclarées dans la PICS (voir le § 9);
- b) choix des groupes de tests qui seront nécessaires pour couvrir de façon appropriée les conditions de conformité et affinage des groupes de tests en ensembles d'objets de test (voir le § 10);
- c) spécification de tests élémentaires génériques pour chaque objet de test, en utilisant une notation de test appropriée (voir le § 11);
- d) choix de la ou des méthodes de tests pour lesquelles les tests élémentaires abstraits complets doivent être spécifiés, et décision des restrictions à imposer aux capacités du testeur inférieur et (si approprié à la ou les méthodes de test choisies) au testeur supérieur et aux procédures de coordination de tests (voir le § 12);
- e) choix de la notation de test utilisée pour spécifier l'ensemble des tests élémentaires abstraits et spécification de tous ces tests élémentaires abstraits, y compris la structuration en modules de tests à utiliser (voir le § 13);
- f) spécification des relations entre les tests élémentaires, ainsi que de celles entre les tests élémentaires et la PICS et, autant que possible, le PIXIT, en sorte de déterminer les restrictions à apporter à la sélection et au paramétrage des tests élémentaires à exécuter et à l'ordre dans lequel ils peuvent être exécutés (voir le § 14);
- g) élaboration de procédures de mise à jour de la suite de tests abstraite (voir le § 15).

La suite de cette section fournit les obligations et directives relatives à chaque étape du processus ci-dessus.

9 Détermination des conditions de conformité et de la PICS**9.1 Introduction**

Avant de spécifier une suite de tests abstraite, son concepteur doit d'abord déterminer quelles sont les conditions de conformité stipulées par la ou les Recommandations* concernées et ce qui doit être déclaré dans le formulaire PICS concernant la mise en œuvre de cette ou ces Recommandations*.

9.2 Conditions de conformité

La section 1 de cette partie de la présente Recommandation spécifie les obligations des concepteurs de protocole, préalables à l'élaboration d'une suite de tests abstraite pour un protocole particulier.

En pratique, les premières Recommandations* OSI* peuvent ne pas contenir de spécifications claires de toutes les conditions de conformité appropriées. En particulier, les conditions de conformité statique peuvent être mal spécifiées ou même omises. Dans de tels cas, le concepteur d'une suite de tests devra contribuer à l'élaboration d'un avenant ou d'un additif à la ou les Recommandations* concernées visant à clarifier les conditions de conformité. Toutefois, si une suite de tests abstraite doit être élaborée avant que les conditions de conformité aient été clarifiées dans la ou les Recommandations* concernées, le concepteur de la suite de tests adoptera la solution à court terme donnée dans l'annexe C et déclarera clairement dans la Recommandation* de cette suite de tests quelles en sont les implications (c'est-à-dire ce qui est supposé être obligatoire; ce qui est supposé être conditionnel et avec quelles conditions; et ce qui est supposé être optionnel).

9.3 *Formulaire PICS*

Si aucun formulaire PICS n'est inclus dans une Recommandation* appropriée de protocole, le concepteur d'une suite de tests fournira un formulaire PICS à ajouter comme additif à cette Recommandation* ou, dans des circonstances exceptionnelles, (comme expliqué au § 7.1.1) comme annexe à la Recommandation* concernant la suite de tests abstraite.

Remarque – Ajouter un additif à une Recommandation* de protocole existante peut être plus rapide que de compléter la Recommandation* concernant la suite de tests abstraite, car le formulaire PICS a des chances d'être beaucoup moins controversé que la suite de tests et, par conséquent, de nécessiter moins de révisions avant de parvenir à un accord sur un texte final.

10 **Structure d'une suite de tests**

10.1 *Obligations de base*

Une suite de tests abstraite comprend un certain nombre de tests élémentaires. Les tests élémentaires sont regroupés en groupes de tests, si nécessaire imbriqués. La structure est hiérarchisée, en ce sens qu'un élément d'un certain niveau sera entièrement contenu dans un élément de niveau plus élevé. Toutefois, la structure n'a pas besoin d'être strictement hiérarchisée, en ce sens que n'importe quel test élémentaire peut se présenter dans plusieurs suites de tests ou groupes de tests. Des groupes de tests similaires peuvent se rencontrer dans plusieurs groupes de tests ou suites de tests de niveaux supérieurs.

Le concepteur d'une suite de tests abstraite doit s'assurer qu'un sous-ensemble des objets des tests de chaque suite de tests abstraite concerne les tests de capacité, et qu'un autre ensemble concerne les tests de comportement. Il n'est pas nécessaire que ceci conduise à des tests élémentaires de comportement et de capacité distincts, car l'utilisation de mêmes modules pour des tests de comportement et pour des tests de capacité n'est pas exclue. Le concepteur de la suite de tests doit donner une explication de la façon dont les objets des tests dérivent de la Recommandation* de protocole ou s'y rapportent. Le concepteur de la suite de tests doit également fournir un résumé précisant la portée de la suite de tests.

10.2 *La structure d'un groupe de tests*

Pour garantir que la suite de tests abstraite résultante couvre de façon adéquate les conditions de conformité concernées, il est conseillé au concepteur de la suite de tests de concevoir, selon une démarche descendante, la structure de cette suite de tests sous la forme de groupes de tests imbriqués (voir la figure 1/X.290, partie 2).

Une même suite de tests peut être structurée en groupes de nombreuses manières; aucune méthode n'est nécessairement bonne et la plus appropriée pour une suite de tests peut ne pas l'être pour une autre. Néanmoins le concepteur d'une suite de tests doit s'assurer que cette suite inclut des tests élémentaires pour chacune des catégories suivantes concernées:

- a) tests de capacité (pour les conditions de conformité statique);
- b) tests de comportement relatifs à un comportement valide;
- c) tests de comportement relatifs à un comportement syntaxiquement non valide;
- d) tests de comportement relatifs à un comportement inopportun;
- e) tests axés sur les PDU envoyées à l'IUT;
- f) tests axés sur les PDU reçues de l'IUT;
- g) tests axés sur les interactions entre ce qui est envoyé et ce qui est reçu;
- h) tests relatifs à chaque caractéristique obligatoire du protocole;
- i) tests relatifs à chaque caractéristique optionnelle qui est réalisée;
- j) tests relatifs à chaque phase de protocole;
- k) variations d'un événement de test se produisant dans un état particulier;
- l) variations de la chronologie et des délais de temporisation;
- m) variations du codage des PDU;
- n) variations des valeurs des paramètres individuels;
- o) variations des valeurs relatives à des combinaisons de paramètres.

Cette liste n'est pas exhaustive; des catégories additionnelles peuvent être nécessaires pour garantir une couverture adéquate des conditions de conformité concernées pour une suite de tests spécifique. En outre, ces catégories se recouvrent en partie et il incombe au concepteur de la suite de tests de les structurer sous une forme hiérarchisée appropriée. La structure suivante est un exemple de suite de tests monocouche, fourni à titre indicatif:

- A Tests de capacité
 - A.1 Caractéristiques obligatoires
 - A.2 Caractéristiques facultatives déclarées par la PICS à satisfaire
- B Tests de comportement – Réponse à un comportement valide de la réalisation homologue
 - B.1 Phase d'établissement de connexion (si pertinent)
 - B.1.1 Axé sur ce qui est envoyé à l'IUT
 - B.1.1.1 Variations des événements de test dans chaque état
 - B.1.1.2 Variations de la chronologie et des délais de temporisation
 - B.1.1.3 Variations du codage
 - B.1.1.4 Variations des valeurs de paramètres individuels
 - B.1.1.5 Combinaison de valeurs de paramètres
 - B.1.2 Axé sur ce qui est reçu de l'IUT
 - structuré comme B.1.1
 - B.1.3 Axé sur les interactions
 - structuré comme B.1.1
 - B.2 Phase de transfert de données
 - structuré comme B.1
 - B.3 Phase de terminaison de connexion (si pertinent)
 - structuré comme B.1
- C Tests de comportement – Réponse à un comportement syntaxiquement non valide de la réalisation homologue
 - C.1 Phase d'établissement de connexion (si pertinent)
 - C.1.1 Axé sur ce qui est envoyé à l'IUT
 - C.1.1.1 Variations des événements de test dans chaque état
 - C.1.1.2 Variations du codage de l'événement non valide
 - C.1.1.3 Variations des valeurs de paramètres individuels non valides
 - C.1.1.4 Variations d'une combinaison de valeurs de paramètres non valides
 - C.1.2 Axé sur ce qu'il est demandé à l'IUT d'envoyer
 - C.1.2.1 Valeurs de paramètres individuels non valides
 - C.1.2.2 Valeurs de combinaisons non valides de paramètres
 - C.2 Phase de transfert de données
 - structuré comme C.1
 - C.3 Phase de terminaison de connexion (si pertinent)
 - structuré comme C.1
- D Tests de comportement – Réponse à des événements inopportuns au niveau de la réalisation homologue
 - D.1 Phase d'établissement de connexion (si pertinent)
 - D.1.1 Axé sur ce qui est envoyé à l'IUT
 - D.1.1.1 Variations des événements de tests dans chaque état
 - D.1.1.2 Variations de la chronologie et des délais de temporisation
 - D.1.1.3 Variations des codages spéciaux
 - D.1.1.4 Variations des valeurs des principaux paramètres individuels
 - D.1.1.5 Variations des valeurs des principales combinaisons de paramètres

D.1.2 Axé sur ce qu'il est demandé à l'IUT d'envoyer

- structuré comme D.1.1

D.2 Phase de transfert de données

- structuré comme D.1

D.3 Phase de terminaison de connexion (si pertinent)

- structuré comme D.1

Si la suite de tests doit couvrir plus d'une couche, une structure de suite de tests monocouche telle que celle qui précède peut être répétée pour chaque couche concernée. En outre, une structure ayant un niveau de détail correspondant peut être élaborée pour les tests de capacités et de comportement de plusieurs couches prises comme un tout, en incluant les interactions entre les activités de couches adjacentes.

10.3 *Objet des tests*

Le concepteur d'une suite de tests doit s'assurer que l'objet de chaque test élémentaire d'une suite de tests abstraite est clairement énoncé. Il est suggéré que l'énoncé des objets des tests soit l'étape suivante de l'affinage de la suite de tests après la définition de sa structure en termes de groupes de tests. Les objets des tests peuvent être établis directement après étude des chapitres de la ou des Recommandations* concernées, qui portent sur les groupes de tests concernés. Pour certains groupes de tests, les objets de ces tests peuvent dériver directement de la table d'états du protocole; pour d'autres, ils pourraient dériver de la définition du codage des PDU ou des descriptions de paramètres particuliers, ou de textes qui spécifient les conditions de conformité pertinentes. Une autre possibilité serait que le concepteur d'une suite de tests emploie une description formelle du ou des protocoles concernés et dérive les objets des tests de cette description au moyen d'une méthode automatisée.

Quelle que soit la méthode utilisée pour dériver les objets des tests, le concepteur d'une suite de tests doit s'assurer qu'il couvre de façon adéquate les conditions de conformité de la ou des Recommandations* concernées. A chaque condition de conformité doit au moins correspondre un objet de test.

Il est en outre possible de donner des directives en ce qui concerne la signification d'une «couverture adéquate» dans le cas de l'exemple ci-dessus. Pour l'exprimer, une notation abrégée est utilisée. La lettre «x» représentera toutes les valeurs appropriées du premier chiffre de l'identificateur du groupe de tests, la lettre «y» le second chiffre: ainsi B.x.y.1 représente 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 et B.3.3.1. Avec cette notation, une «couverture adéquate» minimale pour l'exemple ci-dessus est considérée assurée comme suit:

- a) pour les groupes de tests de capacité (A.1, A.2):
 - 1) au moins un objet de test par caractéristique, classe ou sous-ensemble concerné;
 - 2) au moins un objet de test par type de PDU et par variation importante de chacun d'eux, en utilisant des valeurs «normales» ou par défaut pour chaque paramètre;
- b) pour les groupes de tests concernés par les variations d'événements de test dans chaque état (B.x.y.1, C.x.1.1, D.x.y.1):
 - au moins un événement de test par combinaison d'état ou d'événements concernés;
- c) pour les groupes de tests concernant des temporisateurs et la chronologie (B.x.y.2; D.x.y.2):
 - 1) au moins un objet de test par délai de temporisation défini;
 - 2) au moins un objet de test par réponse très rapide, pour chaque type de PDU concerné;
 - 3) au moins un objet de test par réponse très lente, pour chaque type de PDU concerné;
- d) pour les groupes de tests concernés par les variations de codage (B.x.y.3, C.x.1.2, D.x.y.3):
 - au moins un objet de test par type de variations de codage, pour chaque type de PDU concerné;
- e) pour les groupes de tests concernés par des valeurs de paramètres individuels valides (B.x.y.4, D.x.y.4):
 - 1) pour chaque paramètre entier concerné, des objets de test pour les valeurs limites et une valeur choisie au hasard vers le milieu de la plage;
 - 2) pour chaque paramètre binaire concerné, des objets de test pour autant de valeurs qu'il est praticable, avec au moins toutes les valeurs «normales» ou «courantes»;
 - 3) pour les autres paramètres concernés, au moins un objet de test concerné par une valeur différente de celle qui est considérée comme «normale» ou par défaut dans d'autres groupes de tests;

- f) pour les groupes de tests concernés par des valeurs de paramètres individuels syntaxiquement non valides (C.x.1.3, C.x.2.1):
 - 1) pour chaque paramètre entier concerné, des objets de test pour des valeurs syntaxiquement non valides voisines des valeurs limites autorisées, plus une autre valeur non valide choisie de façon aléatoire;
 - 2) pour chaque paramètre binaire concerné, des objets de test pour autant de valeurs non valides qu'il est praticable;
 - 3) pour tous les autres types de paramètres concernés, au moins un objet de test par paramètre;
- g) pour les groupes de tests concernés par des combinaisons de valeurs de paramètres (B.x.y.5, C.x.1.4, C.x.2.2, D.x.y.5):
 - 1) au moins un objet de test par couple de valeurs «critiques»;
 - 2) au moins un objet de test par couple de paramètres en relation, pour tester une combinaison aléatoire de valeurs pertinentes.

11 Spécification des tests élémentaires génériques

11.1 Introduction

Il est recommandé au concepteur d'une suite de tests de spécifier une suite de tests génériques, en particulier s'il a l'intention d'élaborer plusieurs suites de tests abstraites.

Une suite de tests génériques comprend un test élémentaire générique pour chaque objet de test. Chaque test élémentaire générique sera un affinage de son objet de test, pouvant être utilisé comme racine commune de tests élémentaires abstraits correspondants, pour différentes méthodes de test.

Si une suite de tests génériques est élaborée avant les suites de tests abstraites, cela constituera une étape utile du processus de conception. Si une suite de tests génériques est élaborée après l'élaboration d'au moins une suite de tests abstraite, elle permettra de mettre en relation différentes suites de tests et de rechercher les éventuelles discontinuités de couverture.

11.2 Description des tests élémentaires génériques

Un test élémentaire générique comprend une description textuelle d'un «état initial» (du corps du test) et une spécification du corps du test dans une notation de test normalisée. L'«état initial» n'inclut pas seulement l'état du protocole, mais aussi toutes informations nécessaires concernant l'état du SUT et de l'environnement de test.

Le corps du test est la partie du test élémentaire au cours de laquelle des verdicts concernant l'objet du test sont rendus sur les résultats prévus. Le corps du test:

- a) sera défini dans le style de la méthode de test à distance ou répartie;

Remarque – Les tests élémentaires génériques peuvent utiliser toute la capacité d'expression de ces méthodes (pour de plus amples détails voir le § 12.3.3), y compris des primitives locales abstraites.
- b) rendra des verdicts sur tous les résultats possibles; tous les résultats ayant obtenu le verdict «succès» seront explicitement identifiés, alors que tous les résultats ayant obtenu le verdict «échec» ou «non concluant» seront identifiés ou classés (ce qui peut impliquer une catégorie de verdicts par défaut);
- c) sera décrit en utilisant la notation de test normalisée: la notation combinée arborescente et tabulaire (ou TTCN) définie dans l'annexe D est recommandée.

11.3 Relation entre les tests élémentaires génériques et abstraits

Pour une méthode de test abstraite particulière, beaucoup de tests élémentaires abstraits peuvent être dérivés d'un test élémentaire générique unique. Une grande différence entre un test élémentaire abstrait et un test élémentaire générique, est que le test élémentaire abstrait inclut les spécifications d'un préambule et d'un épilogue. Le préambule démarre à partir d'un état stable choisi et conduit à l'état initial requis du corps du test. L'épilogue démarre à la fin du corps du test et revient dans un état stable choisi.

Le préambule et l'épilogue peuvent être réalisés de différentes façons, selon le niveau de contrôle et d'observation assuré par la méthode de test utilisée, ou la diversité des états stables possibles auxquels le test élémentaire

abstrait dérivé peut commencer ou s'achever. Ces tests élémentaires abstraits sont simplement diverses façons de remplir le même objet de test.

En outre, le corps d'un test élémentaire abstrait peut être différent du test élémentaire générique correspondant, si la méthode de test utilisée pour le test élémentaire abstrait est différente de celle utilisée pour le test élémentaire générique.

Si une suite de tests génériques est élaborée, elle doit être utilisée pour mettre en relation les suites de tests abstraites correspondantes relatives à différentes méthodes de test abstraites.

12 Méthodes de test abstraites

12.1 Introduction

Chaque suite de tests abstraite est spécifiée en termes de contrôle et d'observation d'événements, conformément à une des méthodes de test abstraites définies dans ce chapitre. La méthode de test choisie détermine les points auxquels seront spécifiés le contrôle et l'observation, ainsi que les catégories d'événements qui seront utilisées [par exemple, des $(N - 1)ASP$, $(N)PDU$].

12.2 Spécification générale des méthodes de test abstraites

12.2.1 IUT de systèmes d'extrémité

Pour les IUT de SUT systèmes d'extrémité, quatre catégories de méthodes de test abstraites sont définies: une locale, et trois externes qui supposent que le testeur inférieur est situé à distance du SUT et lui est connecté par une liaison ou un réseau.

En règle générale, les méthodes de test sont décrites par référence à une IUT dans laquelle la couche de plus haut niveau a le numéro « N_t » (couche haute) et dans laquelle le protocole de la couche de plus bas niveau à tester a le numéro « N_b » (couche basse). L'IUT peut mettre en œuvre des protocoles de couches de niveaux inférieurs à « N_b », mais ils ne sont pas concernés par les descriptions de méthodes de test. La description s'applique à des IUT monocouches en prenant N_t égal à N_b .

12.2.2 Primitives locales abstraites

Des concepteurs de suites de tests abstraites peuvent, si nécessaire, définir un ensemble de primitives locales abstraites (ALP) à utiliser pour la spécification de la suite de tests. Une ALP est une description abrégée du contrôle et/ou de l'observation à exercer par le testeur supérieur; ce testeur supérieur ne peut pas être décrit en termes d'ASP, mais provoque les événements ou les changements d'état définis dans la ou les Recommandations* de protocole concernées. Les ALP peuvent être utilisées avec une méthode de test abstraite pour des IUT de systèmes d'extrémité, mais pour simplifier, elles ne seront représentées sur aucune des figures qui illustrent ces méthodes.

Tout test élémentaire qui utilise une ALP figurera à titre optionnel dans la suite de tests abstraite.

12.2.3 Les méthodes de test locales

Abréviation: L (local)

Dans ces méthodes:

- a) la suite de tests abstraite est spécifiée en termes de contrôle et d'observation de $(N_b - 1)ASP$ et de $[(N_b) \text{ à } (N_t)]PDU$;
- b) la suite de tests abstraite est également spécifiée en termes de contrôle et d'observation de $(N_t)ASP$;
- c) la suite de tests abstraite peut également être spécifiée en termes de contrôle et d'observation par le testeur supérieur de primitives locales abstraites (ALP);
- d) ces méthodes nécessitent l'accès aux frontières inférieure et supérieure de l'IUT et la définition d'une correspondance entre les ASP spécifiées et leur réalisation à l'intérieur du SUT;
- e) les spécifications des procédures de coordination des tests sont énoncées dans la suite de tests abstraite, mais aucune supposition n'est faite en ce qui concerne leur réalisation;
- f) les testeurs inférieur et supérieur doivent exercer le contrôle et l'observation des ASP spécifiées et exécuter les procédures de coordination des tests spécifiées. Ils sont supposés intégrer au SUT.

Cette méthode est représentée sur la figure 1/X.290, partie 2.

12.2.4 Méthodes de test externes

12.2.4.1 La méthode de test répartie

Abréviation: D (distributed)

Dans cette méthode:

- a) la suite de tests abstraite est spécifiée en termes de contrôle et d'observation de $(N_b - 1)ASP''$ et de $[(N_b) \text{ à } (N_i)]PDU$;
- b) la suite de tests abstraite est également spécifiée en termes de contrôle et d'observation de $(N_i)ASP$; la méthode nécessite l'accès à la frontière supérieure de l'IUT et la définition d'une correspondance entre les $(N_i)ASP$ et leur réalisation dans le SUT;
- c) la suite de tests abstraite peut également être spécifiée en termes de contrôle et d'observation d'ALP exercés par le testeur supérieur;
- d) les spécifications des procédures de coordination de tests sont énoncées dans la suite de tests abstraite, mais aucune supposition n'est faite quant à leur réalisation;
- e) le testeur supérieur doit exercer le contrôle et l'observation des $(N_i)ASP$ spécifiées et exécuter les effets des procédures de coordination des tests requises; aucune autre supposition n'est faite;
- f) le testeur inférieur doit exercer le contrôle et l'observation des $(N_b)ASP''$ spécifiées, des PDU spécifiées et exécuter les procédures de coordination des tests requises.

Cette méthode est illustrée à la figure 2/X.290, partie 2.

12.2.4.2 La méthode de test coordonnée

Abréviation: C (coordinated)

Dans cette méthode:

- a) la suite de tests abstraite est spécifiée en termes de contrôle et d'observation de $(N_b - 1)ASP''$, de $[(N_b) \text{ à } (N_i)]PDU$ et de PDU de gestion de tests (TMPDU);
- b) des $(N_i)ASP$ doivent être utilisées dans la spécification de la suite de tests abstraite; aucune hypothèse n'est faite quant à l'existence d'une frontière supérieure de l'IUT;
- c) les spécifications des procédures de coordination de tests sont énoncées dans la suite de tests abstraite, au moyen d'un protocole de gestion de tests normalisé;
- d) des TMPDU qui correspondent aux ALP peuvent être définies;
- e) le testeur supérieur doit mettre en oeuvre le protocole de gestion de tests et exercer les effets appropriés sur l'IUT;
- f) le testeur inférieur doit exercer le contrôle et l'observation des $(N_b)ASP''$ spécifiées et des PDU spécifiées (y compris les TMPDU).

Cette méthode est illustrée à la figure 3/X.290, partie 2.

12.2.4.3 La méthode de test à distance

Abréviation: R (remote)

Cette méthode est définie pour le cas où il n'est pas possible d'observer et de contrôler la frontière supérieure de l'IUT. Ainsi, dans cette méthode:

- a) la suite de tests abstraite est spécifiée en termes de contrôle et d'observation de $(N_b - 1)ASP''$ et de $[(N_b) \text{ à } (N_i)]PDU$;
- b) les $(N_i)ASP$ ne sont pas utilisées dans la spécification de la suite de tests abstraite; aucune hypothèse quant à l'existence d'une frontière supérieure de l'IUT;
- c) la suite de tests abstraite peut également être décrite en termes de contrôle et d'observation des ALP échangées à l'intérieur du SUT;
- d) certaines spécifications des procédures de coordination de tests peuvent être implicites, ou exprimées de façon informelle dans la suite de tests abstraite mais aucune supposition n'est faite concernant leur faisabilité ou leur réalisation;
- e) d'un point de vue abstrait, le SUT doit exécuter certaines fonctions du testeur supérieur, pour exercer tous les effets des procédures de coordination de tests, et tous les contrôles et/ou observations de l'IUT impliqués, qu'ils soient décrits de façon informelle ou par des ALP, dans la suite de tests abstraite d'un

protocole donné; ces fonctions ne sont pas spécifiées et aucune supposition n'est faite concernant leur faisabilité ou leur réalisation;

- f) le testeur inférieur doit exercer le contrôle et l'observation des $(N_b)ASP''$ et des PDU spécifiées, et doit essayer d'exécuter les procédures de coordination des tests implicites ou exprimées de façon informelle, conformément aux informations pertinentes du PIXIT.

Cette méthode est illustrée à la figure 4/X.290, partie 2.

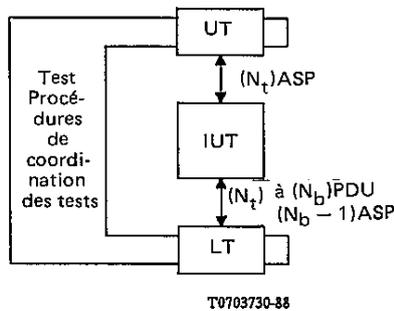


FIGURE 1/X.290, Partie 2

La méthode de test locale

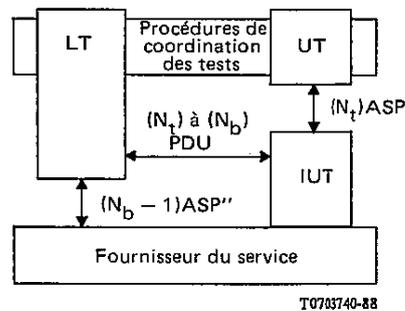


FIGURE 2/X.290, Partie 2

La méthode de test répartie

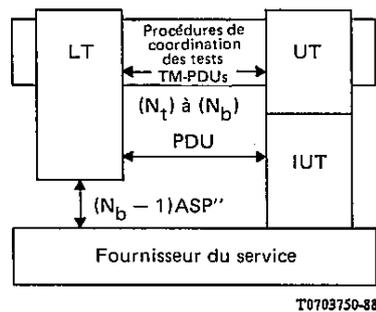


FIGURE 3/X.290, Partie 2

La méthode de test coordonnée

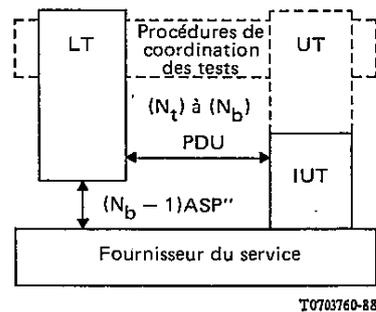


FIGURE 4/X.290, Partie 2

La méthode de test à distance

12.2.5 Variantes monocouches, multicouches et encastrées

Chaque catégorie de méthodes de test a une variante applicable à une IUT monocouche (abréviation: S); et une autre applicable aux IUT multicouches (abréviation: M), quand l'ensemble des couches adjacentes doit être testé sous forme combinée (comme un tout).

Une variante encastrée des méthodes de test a été définie pour une IUT multicouche dans laquelle les protocoles doivent être testés couche par couche (abréviation: E).

Si le contrôle et l'observation sont exercés au moyen d'un accès à la frontière supérieure des entités à tester à l'intérieur du SUT, les méthodes de test sont normales (et la lettre E n'est pas ajoutée au nom abrégé). En revanche, si le contrôle et l'observation sont exercés au travers d'entités d'une ou plusieurs couches OSI* situées au-dessus des entités à tester, les méthodes de test sont appelées encastrées (et la lettre E est ajoutée à la fin de l'abréviation).

Les noms des diverses variantes des méthodes de test seront formés comme suit:

$$\begin{array}{|c|} \hline L \\ \hline R \\ \hline C \\ \hline D \\ \hline \end{array} \quad \begin{array}{|c|} \hline S \\ \hline M \\ \hline \end{array} \quad [E]$$

Par exemple, DSE est l'abréviation de méthode de test «monocouche encastrée répartie».

12.2.6 *Systèmes relais ouverts*

Pour les systèmes ouverts relais, des méthodes de test abstraites en boucle et transversale sont définies. Leurs abréviations sont respectivement YL et YT.

12.3 *Méthodes de test monocouches pour des IUT monocouches de systèmes d'extrémité*

Pour les méthodes de test monocouches, le modèle abstrait de l'IUT est appelé entité (N) à tester.

12.3.1 *La méthode de test monocouche locale*

Dans la méthode de test abstraite monocouche locale (LS) les points de contrôle et d'observation sont, par définition, aux frontières de service situées au-dessus et en dessous de l'entité (N) à tester. Les événements de test sont spécifiés en termes de (N)APS échangées au-dessus de l'IUT et de (N – 1)APS et (N)PDU échangées en dessous de l'IUT, comme représenté à la figure 5/X.290, partie 2. En outre, les ALP peuvent être utilisées comme événements de test. D'un point de vue abstrait, un testeur inférieur est considéré comme observant et contrôlant les (N – 1)ASP et les (N)PDU alors qu'un testeur supérieur observe et contrôle les (N)ASP et les ALP. Les conditions à remplir par les procédures de coordination de tests utilisées pour coordonner des réalisations des testeurs supérieur et inférieur sont définies dans les suites de tests abstraites, alors que les procédures de coordination de tests ne le sont pas.

12.3.2 *La méthode de test monocouche répartie*

Dans la méthode de test abstraite monocouche répartie (DS) les points de contrôle et d'observation sont, par définition, aux frontières de service situées au-dessus de l'entité (N) à tester et en dessous du fournisseur du service (N - 1) au SAP distant de l'entité (N) à tester. Les événements de test sont spécifiés en termes de (N)ASP échangées au-dessus de l'IUT et de (N – 1)ASP" et de (N)PDU échangées à distance, comme représenté sur la figure 6/X.290, partie 2. En outre, les ALP peuvent être utilisées comme événements de test. D'un point de vue abstrait, les testeurs inférieur et supérieur sont considérés observer et contrôler le comportement à leurs PCO respectifs. Les conditions à remplir par les procédures de coordination de tests sont également définies en termes de suites de tests abstraites, alors que les procédures elles-mêmes ne le sont pas.

Pour les couches de niveaux inférieurs (1 à 3) dans lesquelles il serait irréaliste de spécifier l'observation et le contrôle de (N – 1)ASP", l'observation et le contrôle du testeur inférieur seront spécifiés en termes de (N)PDU et, si nécessaire, de modifications d'état de la connexion sous-jacente.

Remarque – Par exemple, l'état de la connexion sous-jacente peut être modifié en établissant une nouvelle connexion ou en rétablissant ou en fermant une connexion existante.

L'observation et le contrôle à exercer par le testeur inférieur peuvent, facultativement, être spécifiés en termes de (N)APS" lorsque ceci permet de réduire la taille de la spécification du test élémentaire sans réduire la précision requise.

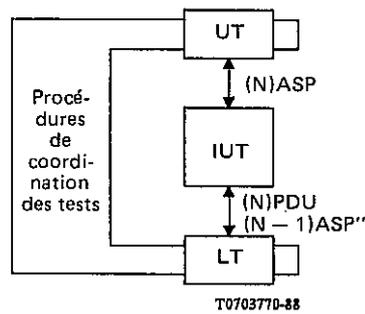


FIGURE 5/X.290, Partie 2

La méthode de test LS

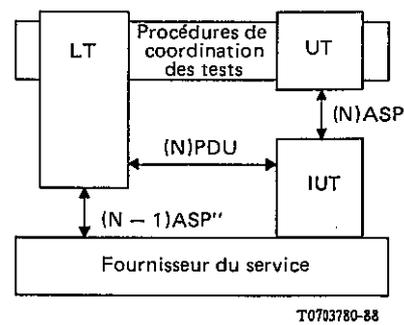


FIGURE 6/X.290, Partie 2

La méthode de test DS

12.3.3 La méthode de test monocouche coordonnée

La méthode de test abstraite monocouche coordonnée (CS) est une version améliorée de la méthode DS, qui utilise un testeur supérieur normalisé et la définition d'un protocole de gestion de tests pour réaliser les procédures de coordination de tests entre les testeurs supérieur et inférieur. Les mêmes testeurs supérieurs et protocoles de gestion de tests normalisés ne sont pas nécessairement applicables à toutes les suites de tests qui utilisent la méthode coordonnée.

Avec la méthode de test coordonnée, les testeurs supérieurs et les protocoles de gestion de tests normalisés sont applicables à une suite de tests abstraite normalisée particulière et peuvent ne pas l'être à d'autres suites de tests abstraites.

Un seul point de contrôle et d'observation est défini au-dessus du fournisseur de service $(N - 1)$ au SAP distant de l'entité (N) à tester. Les événements de test sont spécifiés en termes de $(N - 1)ASP''$, de $(N)PDU$ et de $TMPDU$, comme représenté sur la figure 7/X.290, partie 2.

Pour les couches de niveaux inférieurs (1 à 3), dans lesquelles il serait irréaliste de spécifier l'observation et le contrôle de $(N - 1)ASP''$, l'observation et le contrôle seront spécifiés en termes de $TMPDU$, $(N)PDU$ et, si nécessaire, de modifications d'état de la connexion sous-jacente.

En ce qui concerne le protocole de gestion de tests:

- le protocole de gestion de tests doit être mis en oeuvre dans le SUT se trouvant immédiatement au-dessus de la frontière du service abstrait située au-dessus de l'IUT;
- il ne sera pas imposé à l'IUT d'interpréter les $TMPDU$, mais seulement de les transmettre au testeur supérieur et d'en recevoir de celui-ci;
- un protocole de gestion de tests est seulement défini pour le test d'un protocole particulier et n'a donc pas besoin d'être indépendant du protocole sous-jacent;
- les verdicts rendus sur les tests élémentaires ne seront pas fondés sur l'aptitude du SUT à présenter telle ASP ou tel paramètre d'une ASP à la frontière de service supérieure de l'unité, car ceci serait en contradiction avec la définition de la méthode coordonnée, en ce sens que la frontière de service supérieure de l'IUT n'est pas un PCO dans cette méthode. Toutefois, il est recommandé de définir le protocole de gestion de tests séparément de la ou des suites de tests abstraites, pour faciliter la tâche du réalisateur d'un testeur supérieur. Cette définition (de même que la définition de tout protocole OSI* défini par l'ISO ou le CCITT) peut se référer à des ASP de son service sous-jacent (c'est-à-dire les ASP de la frontière de service supérieure de l'IUT);
- les $TMPDU$ qui correspondent aux ALP sont optionnelles et le testeur supérieur n'est pas obligé de les prendre en charge.

12.3.4 La méthode de test monocouche à distance

Dans la méthode de test abstraite monocouche à distance (RS), le point de contrôle et d'observation est, par définition, au-dessus du fournisseur de service $(N - 1)$ au SAP distant de l'entité (N) à tester. Les événements de service sont spécifiés en termes de $(N - 1)ASP''$ et de $(N)PDU$ échangées à distance, comme représenté sur la figure 8/X.290, partie 2. En outre, des ALP peuvent être utilisées comme événements de test. Certaines spécifications des procédures de

coordination de tests peuvent être implicites ou exprimées de façon informelle dans les suites de tests abstraites, mais aucune hypothèse ne sera faite quant à leur faisabilité ou leur réalisation.

Pour les couches des niveaux inférieurs (1 à 3), où il serait irréaliste de spécifier l'observation et le contrôle de (N - 1)ASP'', l'observation et le contrôle seront spécifiés en termes de (N)PDU et, si nécessaire, de changements d'état de la connexion sous-jacente.

En outre, pour pallier l'absence de spécifications du comportement au-dessus de l'entité (N) à tester, le comportement requis du système à tester sera spécifié, si nécessaire, en termes de (N - 1)ASP'' ou de (N)PDU qui doivent être observés par le testeur inférieur. Cette forme de spécification implicite sera considérée comme signifiant «faire tout ce qui est nécessaire dans le système à tester pour provoquer ce comportement requis».

Remarque – Une telle spécification implicite est nécessaire avec cette méthode de test pour tout test élémentaire qui nécessite un événement déclenché par l'IUT et qui ne peut pas l'être au moyen d'une ALP. Comme les ALP ne peuvent être définies que si le même effet ne peut pas être obtenu par des ASP, dans cette méthode de test, toute PDU dont l'émission peut être provoquée par une ASP doit être spécifiée de façon implicite pour pouvoir être émise.

Toutefois, il est possible que certains tests élémentaires de la suite de tests abstraite ne puissent pas être exécutés (par exemple la transmission et le maintien d'états d'occupation, la transmission de PDU de données n'ayant pas fait l'objet d'un accusé de réception, etc.). Dans de tels cas, il incombe au laboratoire de test et au client de négocier la méthode par laquelle ces tests peuvent être réalisés.

Même avec une telle spécification implicite du contrôle et de l'observation de l'IUT, il est possible avec cette méthode de spécifier le contrôle au-dessus de l'IUT, mais pas l'observation, sauf en utilisant des ALP. C'est une différence essentielle entre la méthode de test RS et les méthodes DS et CS.

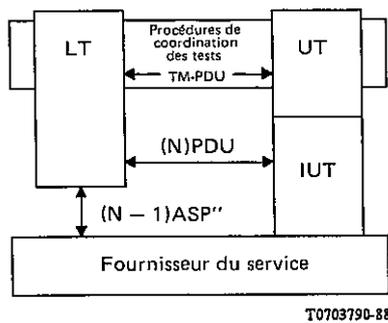


FIGURE 7/X.290, Partie 2
La méthode de test CS

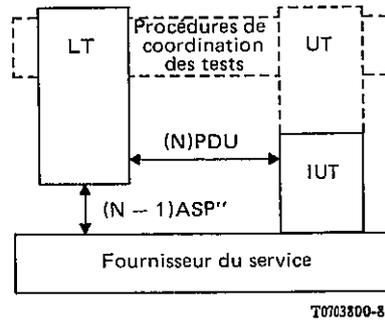


FIGURE 8/X.290, Partie 2
La méthode de test RS

12.4 Méthodes de test multicouches pour des IUT multicouches (LM, DM, CM, RM)

Le test multicouche, quand le comportement autorisé combiné de la réalisation multicouche est connu, comprend le test de toutes les couches de l'IUT multicouche, prise comme un tout, sans contrôle ni observation d'aucune des frontières intercouches intérieures à l'IUT.

Dans la méthode de test multicouche locale (LM), les points d'observation et de contrôle sont les frontières de service situées immédiatement au-dessus et en dessous de l'IUT. Les événements de test peuvent être spécifiés en fonction de (N_i)ASP et d'ALP échangées au-dessus de l'IUT et de (N - 1)ASP et de (N)PDU et (N_i)PDU échangées en dessous de l'IUT.

Dans la méthode de test multicouche répartie (DM), les points d'observation et de contrôle sont à la frontière de service située au-dessus de l'IUT et au-dessus du fournisseur du service (N - 1) au SAP distant de l'IUT. Les événements de test peuvent être spécifiés en fonction de (N_i)ASP et d'ALP échangées au-dessus de l'IUT et de (N - 1)ASP'' et de (N)PDU et (N_i)PDU échangées à distance.

Dans la méthode de test multicouche coordonnée (CM), le point d'observation et de contrôle est au-dessus du fournisseur du service (N - 1) au SAP distant de l'IUT. Les événements de test seront spécifiés en fonction de (N - 1)ASP'', de (N)PDU et (N_i)PDU, et de TMPDU. Le protocole de gestion de tests sera conçu pour fonctionner au-dessus du service (N_i), où (N_i) est la couche de plus haut niveau de l'IUT.

Dans la méthode de test multicouche à distance (RM), le point d'observation et de contrôle est au-dessus du fournisseur du service ($N - 1$) au SAP distant de l'IUT. Les événements de test seront spécifiés en fonction de $(N - 1)ASP''$, de $[(N) \text{ à } (N_i)]PDU$, d'ALP et de la spécification implicite du contrôle du comportement du SUT si nécessaire. Certaines spécifications des procédures de coordination de tests peuvent être implicites ou exprimées de façon informelle, mais aucune supposition ne sera faite quant à leur faisabilité ou leur réalisation.

12.5 *Test multicouche d'IUT ou de SUT multicouches (méthodes encastrées)*

Dans les méthodes de test monocouches encastrées, les tests sont spécifiés pour une seule couche d'une IUT multicouche, en incluant la spécification de l'activité du protocole dans les couches situées au-dessus de celle qui est testée, mais sans spécifier le contrôle et l'observation aux frontières de service à l'intérieur de cette IUT multicouche. Ainsi, dans une IUT multicouche allant de la couche (N) à (N_i) , des tests élémentaires abstraits portant sur la couche (N_i) incluront la spécification des PDU des couches $(N_i + 1)$ à (N_i) ainsi que celle de la couche (N_i) .

La méthode de test monocouche encastrée locale (LSE) utilise les mêmes points de contrôle et d'observation que la méthode de test LM pour le même ensemble de couches. Les événements de test seront également spécifiés dans les mêmes termes que pour cette méthode. La différence est que la méthode de test LSE est concentrée sur une seule couche à la fois, alors que la méthode de test LM porte sur une IUT multicouche prise comme un tout.

Dans la méthode de test monocouche encastrée répartie (DSE), appliquée à la couche (N_i) d'une IUT multicouche de la couche (N) à la couche (N_i) , les points d'observation et de contrôle sont à la frontière de service située au-dessus de l'IUT et au-dessus du fournisseur du service $(N_i - 1)$ au SAP distant de l'IUT, comme représenté dans la figure 9a)/X.290, partie 2. Les événements de test seront spécifiés en fonction de $(N_i)ASP$ et d'ALP échangées au-dessus de l'IUT et de $(N_i - 1)ASP''$ et $[(N_i) \text{ à } (N_i)]PDU$ échangées à distance.

Remarque – Pour la couche supérieure (N_i) d'une IUT multicouche, cette méthode est identique à la méthode de test DS.

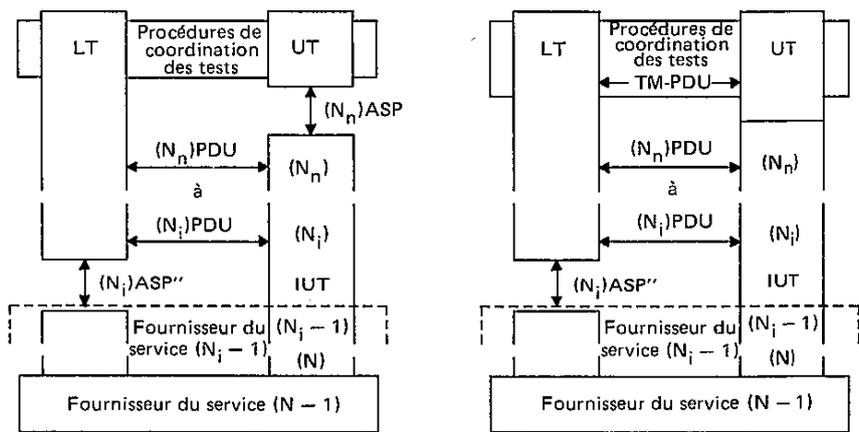
La méthode de test monocouche encastrée coordonnée (CSE) utilise des caractéristiques des méthodes de test CM et DSE. Les événements de test seront spécifiés en termes de $(N_i - 1)ASP''$, de $[(N_i) \text{ à } (N_i)]PDU$ et de TMPDU, et le protocole de gestion de tests sera conçu pour fonctionner au-dessus du service (N_i) . Ceci est représenté dans la figure 9b)/X.290, partie 2.

La méthode de test monocouche encastrée à distance (RSE) utilise le même point de contrôle et d'observation que la méthode RS pour la même couche, mais diffère de cette méthode en ce que des $[(N_i + 1) \text{ à } (N_i)]PDU$ seront spécifiées dans les tests élémentaires de la couche (N_i) .

L'utilisation successive d'une méthode de test monocouche encastrée [de la couche (N) à la couche (N_i)] est appelée test incrémentiel d'une IUT multicouche.

Les méthodes DSE, CSE et RSE sont définies pour une couche unique à tester dans une IUT multicouche. Ceci ne signifie pas qu'il n'existe pas de frontières de service accessibles à l'intérieur de cette IUT multicouche, mais seulement que ces frontières ne sont pas utilisées dans les méthodes de test. Ainsi, toutes les couches comprises entre la couche à tester et la couche haute, pour lesquelles les PDU sont exprimées en événements de test d'une suite de tests abstraite, seront considérées comme faisant partie de l'IUT multicouche.

Remarque – Les méthodes de test DME, CME et RME pourraient théoriquement être définies pour s'appliquer à plusieurs couches prises comme un tout dans une IUT multicouche plus importante, mais pour éviter une complexité excessive, ceci n'est pas détaillé dans la présente Recommandation.



a) Exemple de méthode de test DSE:
Test d'un protocole (N_i) dans une IUT
de couches (N) à (N_n)

b) Exemple de méthode de test CES:
Test d'un protocole (N_i) dans une IUT
de couches (N) à (N_n)

FIGURE 9/X.290, Partie 2
Les méthodes de test encastrées

12.6 Méthodes de test relais

Deux méthodes de test abstraites sont définies pour les systèmes relais:

- a) La méthode de test en boucle (YL): utilisée pour tester un système relais à partir d'un sous-réseau.

Cette méthode de test est représentée sur la figure 10/X.290, partie 2.

Elle comporte deux points d'observation et de contrôle sur un sous-réseau, au SAP distant du relais (N) . Pour les protocoles en mode connexion, elle nécessite que les deux connexions de test soient bouclées ensemble du côté distant du relais (N) , mais il n'est pas spécifié si ce bouclage est effectué à l'intérieur du relais (N) ou dans le second sous-réseau. Pour les protocoles en mode sans connexion, elle nécessite que les PDU soient rebouclées à l'intérieur du second sous-réseau et adressées en retour au second point d'observation et de contrôle.

- b) La méthode de test transversale (YT): utilisée pour tester un système relais à partir de deux sous-réseaux.

Cette méthode de test est représentée sur la figure 11/X.290, partie 2.

Cette méthode de test comporte deux points d'observation et de contrôle, un sur chaque sous-réseau, aux SAP distants du relais (N) .

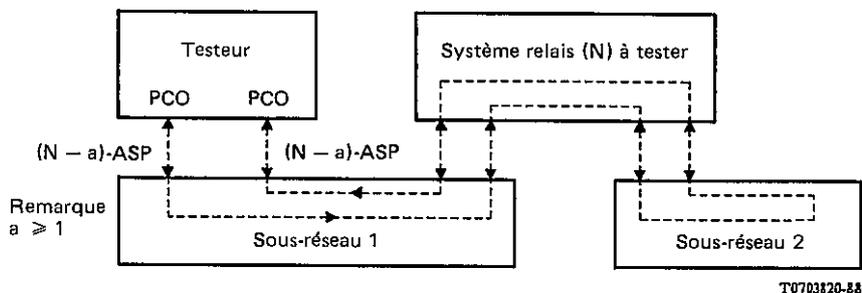


FIGURE 10/X.290, Partie 2
Méthode de test en boucle (YL)

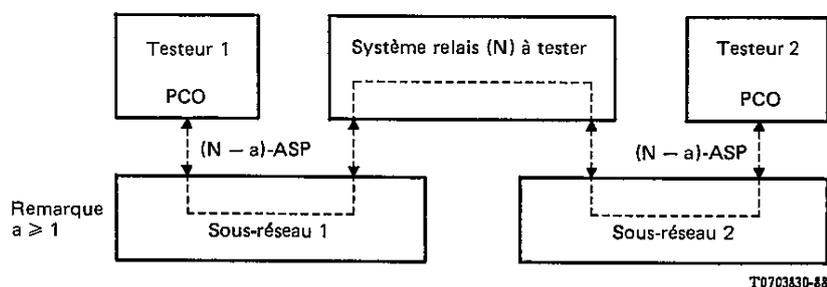


FIGURE 11/X.290, Partie 2
Méthode de test transversale (YT)

12.7 Choix de la méthode de test

12.7.1 Introduction

Pour pouvoir définir une suite de tests abstraite, il est nécessaire d'étudier tous les environnements dans lesquels le protocole est susceptible d'être testé et de définir en conséquence la ou les méthodes de test abstraites à utiliser pour l'élaboration d'une ou plusieurs suites de tests abstraites.

Les concepteurs de suites de tests abstraites doivent inclure une condition dans la Recommandation* concernant des suites de tests abstraites, spécifiant quelle ou quelles méthodes de test abstraites doivent être assurées au minimum par un organisme déclarant fournir un service de test complet pour le ou les protocoles en question.

12.7.2 Environnements des IUT

Il existe une relation entre les méthodes de test et les configurations des systèmes ouverts réels à tester.

Le § 7.2 de la partie 1 de la présente Recommandation présente une classification complète des systèmes et des IUT.

Pour choisir une méthode de test, les concepteurs de suites de tests doivent déterminer, s'ils ne l'ont pas déjà fait, si les suites de tests qu'ils prévoient d'élaborer concernent des IUT qui:

- sont monocouches ou multicouches;
- appartiennent à un système d'extrémité ou à un système relais;
- appartiennent à un système complet ou partiel;
- appartiennent à un système intégralement ouvert ou mixte;
- ont des frontières de service accessibles ou non;
- sont à usage spécifique (c'est-à-dire à utiliser pour une application unique) ou à usage général (c'est-à-dire à utiliser pour plusieurs applications).

12.7.3 Applicabilité des méthodes de test abstraites

La possibilité d'élaborer une suite de tests abstraite pour une méthode de test abstraite donnée dépendra du ou des protocoles considérés, ainsi que des résultats de l'étude décrite au § 12.7.2. Ceci s'applique à la possibilité d'élaborer des suites de tests pour une combinaison donnée de méthodes (par exemple incrémentielles) ou pour une variante donnée d'une méthode (par exemple encastrée).

Certaines considérations concernant l'applicabilité des méthodes à différentes couches sont présentées dans l'appendice I de la partie 1 de la présente Recommandation.

Une ou plusieurs méthodes de test abstraites appropriées seront choisies pour le protocole considéré.

Des priorités seront affectées aux diverses méthodes de test qui ont été déterminées, en fonction des configurations les plus probables dans les systèmes réels.

12.7.4 Exemples

L'appendice III donne un exemple illustrant le choix des méthodes de test abstraites pour des protocoles donnés.

12.8 Procédures de coordination de tests

Pour une exécution efficace et fiable des tests de conformité, un certain ensemble de règles est requis pour la coordination du processus de test entre le testeur inférieur et le testeur supérieur. L'objectif général de ces règles est de permettre au testeur inférieur de contrôler à distance le fonctionnement du testeur supérieur, ou vice versa, selon des modalités nécessaires à l'exécution de la suite de tests choisie pour l'IUT.

Ces règles conduisent à l'élaboration de procédures de coordination de tests visant à réaliser la synchronisation entre le testeur inférieur et le testeur supérieur, et la gestion des informations échangées durant le processus de test. Les détails de ces procédures et la façon dont ces effets sont obtenus dépendent étroitement des caractéristiques du SUT ainsi que des méthodes de test externes.

Un ensemble normalisé de procédures de coordination de tests sera élaboré pour chaque suite de tests abstraite utilisant les méthodes de test coordonnées, réparties ou locales. Ces procédures dépendent en effet des fonctions du testeur supérieur et des définitions des tests élémentaires.

Dans le cas des méthodes de test coordonnées (CS, CSE, CM), les procédures de coordination de tests sont réalisées par la normalisation du protocole de gestion de tests. Le protocole de gestion de tests doit être capable de véhiculer des demandes adressées à l'IUT pour avoir l'effet d'une primitive de service ou d'une ALP et de les véhiculer en sens inverse vers le testeur inférieur pour enregistrer les observations des faits équivalant à l'occurrence de primitives de service ou d'ALP. Le testeur supérieur sera une réalisation du protocole de gestion de tests. Il sera nécessaire d'ajouter des tests élémentaires à la suite de tests abstraite pour vérifier que le testeur supérieur est conforme aux conditions de la spécification du protocole de gestion de tests. De tels tests élémentaires ne contribuent pas à l'évaluation de conformité de l'IUT.

Lors de la définition de tests élémentaires de méthodes de test locales et réparties, le concepteur d'une suite de tests consignera toutes les contraintes imposées au testeur supérieur et/ou concernant les procédures de coordination de tests qui peuvent être nécessaires.

13 Spécification des suites de tests abstraites

13.1 Tests élémentaires

13.1.1 Le concepteur d'une suite de tests abstraite doit choisir une notation normalisée appropriée pour définir les tests élémentaires abstraits. La notation combinée arborescente et tabulaire (TTCN), définie dans l'annexe D, est spécifiée à cette fin.

13.1.2 Quand la notation de test et la méthode de test auront été choisies, les tests élémentaires génériques pourront être développés en tests élémentaires abstraits. Deux importantes modifications sont nécessaires pour convertir un test élémentaire générique en test élémentaire abstrait. La première est d'exprimer le corps du test en termes de contrôle et d'observation requis par la méthode de test et, le cas échéant, d'inclure une description de la synchronisation nécessaire entre les testeurs supérieur et inférieur. La seconde modification importante est de spécifier le préambule et l'épilogue.

13.1.3 La spécification du préambule et de l'épilogue peut, pour chacun, se traduire par plusieurs modules de tests. Le préambule devra démarrer dans un état stable et évoluer vers l'état désiré. Réciproquement, l'épilogue évoluera de l'état final du corps du test vers un état stable. Un nombre réduit d'états stables sera défini pour le protocole concerné, contenant toujours au minimum l'état «repos». Chaque test élémentaire devra être capable de s'exécuter de façon autonome et inclura donc les étapes de test de lancement du préambule à partir de l'état de «repos», et de terminaison de l'épilogue dans l'état «repos».

Toutefois, d'autres états stables de lancement et de terminaison d'un test élémentaire abstrait peuvent être utilisés quand est requise une concaténation efficace des tests élémentaires abstraits.

En outre, le concepteur d'une suite de tests peut avoir avantage, pour la structuration des modules de tests d'une suite de tests abstraite, à utiliser les mêmes modules pour plusieurs tests élémentaires abstraits.

13.1.4 Lors de la conversion des tests élémentaires génériques en tests élémentaires abstraits, le concepteur d'une suite de tests doit s'assurer que l'état initial du corps du test est préservé, que les itinéraires au travers du corps du test sont préservés et que les affectations de verdicts aux résultats restent cohérentes.

Pour préserver la cohérence, les règles conditionnelles suivantes s'appliquent aux verdicts rendus sur les résultats:

- a) si le comportement du préambule et de l'épilogue est valide, le verdict rendu sur un résultat particulier sera identique à celui rendu sur le résultat correspondant du test élémentaire générique;
- b) si le préambule ne permet pas d'atteindre l'état initial du corps du test, le verdict sera «non concluant», même si ce n'est pas du fait d'un comportement non valide;
- c) si le préambule ne permet pas d'atteindre l'état initial du corps du test, du fait d'un comportement non valide, alors le verdict sera «échec mais objet du test non concluant» («échec de type 3»);
- d) si l'épilogue présente un comportement non valide et que le verdict du test élémentaire générique (ou du corps du test) est «succès», ou «non concluant», le verdict sera respectivement «échec mais objet du test non rempli» («échec de type 2») ou «échec mais objet du test non concluant» («échec de type 3»);
- e) si le verdict du test élémentaire générique (ou corps du test) est «échec», un comportement non valide dans l'épilogue ne modifiera pas le verdict («échec de type 1»).

13.1.5 Le concepteur d'une suite de tests doit également vérifier que chaque test élémentaire abstrait identifie explicitement tous les résultats pour lesquels est rendu le verdict «succès» et identifie ou classe les résultats prévus restants, rendant pour chaque résultat individuel ou catégorie de résultats un verdict «échec» ou «non concluant». Pour tous les résultats imprévus du corps du test, le verdict rendu doit être l'un des suivants:

- a) «échec»;
- b) «non concluant».

Le concepteur de la suite de tests doit vérifier que l'application des verdicts a) ou b) est cohérente d'un bout à l'autre de la suite de tests abstraite. Si a) est choisi pour tout résultat imprévu du préambule, le verdict rendu sera «échec mais objet du test non concluant» («échec de type 3») et tout résultat imprévu de l'épilogue doit être traité comme une transgression du protocole, conduisant au type approprié de verdict d'échec.

Si b) est choisi, le verdict rendu pour tout résultat imprévu du préambule doit être «échec mais objet du test non concluant» («échec de type 3») et tout résultat imprévu de l'épilogue doit donner lieu au type approprié de verdict d'échec.

13.2 Suites de tests

Une spécification de suite de tests abstraite comprend un ensemble de tests élémentaires et de modules de tests. Les informations suivantes doivent précéder les tests élémentaires proprement dits:

- a) nom de la suite de tests abstraite, date d'origine et numéro de version;
- b) noms (et numéros de version) de la ou des Recommandations* de protocole pour lesquelles des tests élémentaires sont fournis;
- c) noms (et numéros de version) de la ou des Recommandations* de service dont les primitives sont spécifiées comme observées;
- d) nom (et numéro de version) de la Recommandation* définissant la notation de test, ou référence à une annexe de cette Recommandation* si la notation n'est pas normalisée ailleurs;
- e) nom de la méthode de test cible;
- f) description de la couverture assurée par la suite de tests; par exemple, les sous-ensembles fonctionnels de la ou des Recommandations* de protocole qui sont testés;
- g) description de la structure de la suite de tests, en termes de groupes de tests et de leurs relations avec la ou les Recommandations* de protocole;
- h) description des procédures de coordination de tests (si applicable dans la méthode de test);
- i) déclaration des tests élémentaires qui sont facultatifs et de ceux qui sont obligatoires pour la conformité à la Recommandation* de suite de tests abstraite;
- j) informations à l'intention du réalisateur de l'équipement de test et du laboratoire de test, pour les aider à utiliser la Recommandation* de suite de tests abstraite (voir le § 14).

14 Utilisation d'une spécification de suite de tests abstraite

Le concepteur d'une suite de tests abstraite fournira des informations dans la Recommandation* de suite de tests abstraite pour aider le réalisateur de l'équipement de test et le laboratoire de test à utiliser la suite de tests. Ces informations incluront, à titre non limitatif, les éléments suivants:

- a) la définition d'une correspondance entre les tests élémentaires abstraits et les entrées du formulaire PICS, permettant de déterminer si chaque test élémentaire abstrait doit ou non être choisi pour une certaine IUT; la correspondance sera spécifiée selon une notation appropriée aux expressions booléennes;
- b) la définition d'une correspondance entre les tests élémentaires abstraits et les entrées du formulaire PIXIT, dans la mesure où elles sont connues du concepteur de la suite de tests abstraite, permettant de paramétrer la suite de tests pour une IUT particulière, et de déterminer quels tests élémentaires sélectionnés ne peuvent pas être exécutés sur cette IUT particulière;

Le concepteur de la suite de tests abstraite définira un formulaire PIXIT partiel. Il contiendra une liste de toutes les ALP utilisées dans la suite de tests (ou dans le protocole de gestion de tests) et une liste de tous les paramètres pour lesquels la suite de tests nécessite des valeurs. Si des valeurs de paramètres requises se trouvent dans la PICS, l'entrée correspondante du formulaire PIXIT fera référence à l'entrée correspondante du formulaire PICS;

Remarque – D'autres aspects du formulaire PIXIT doivent faire l'objet d'un complément d'étude;

- c) une liste des tests élémentaires abstraits, dans l'ordre qui sera utilisé dans le rapport de test de conformité au protocole (PCTR) ainsi que toutes informations qui auront été conservées dans le PCTR sur l'état de chaque test élémentaire; ceci afin de contribuer à l'élaboration d'un formulaire PCTR;
- d) toutes restrictions pouvant être formulées sur l'ordre dans lequel les tests élémentaires peuvent être exécutés;
- e) une référence à la description des procédures de coordination de test (si applicable dans la méthode de test choisie);
- f) toutes les informations chronologiques nécessaires.

15 Mise à jour des suites de tests

Une fois qu'une suite de tests a été spécifiée et mise en service, on peut s'attendre à la détection d'erreurs et d'omissions par ceux qui l'utilisent. Dans de telles circonstances, le concepteur d'une suite de tests abstraite doit procéder à sa mise à jour, en appliquant les procédures d'amendement rapide appropriées.

En outre, des modifications seront apportées périodiquement à la ou aux Recommandations* de protocole qui concernent la suite de tests. Le concepteur de la suite de tests abstraite devra s'assurer qu'elle est mise à jour le plus tôt possible après la ratification des modifications à apporter à la ou aux Recommandations* de protocole concernées.

ANNEXE A

(à la Recommandation X.290, partie 1)

Options

A.1 Les options sont les éléments d'une Recommandation* parmi lesquels le réalisateur peut choisir ceux qui conviennent à sa réalisation.

A.2 Ce choix n'est pas entièrement libre. Des spécifications stipulent les conditions d'application des options et les limitations de ce choix.

Réciproquement, des conditions obligatoires ou conditionnelles, ou des interdictions stipulées par une Recommandation* peuvent dépendre d'un choix ou d'une combinaison des choix déjà opérés.

A.3 Voici des exemples d'options et de conditions associées; la liste n'est pas exhaustive:

- a) options «booléennes»: l'option est «faire ou ne pas faire»; l'obligation est «dans l'affirmative, faire comme spécifié»;
- b) options mutuellement exclusives: l'obligation est de faire exactement une action parmi n ; l'option est quelle action faire;

- c) options sélectives: l'option est de faire m actions parmi n , avec l'obligation de faire au moins une action ($1 \leq m \leq n$ et $n \geq 2$).

A.4 Les options peuvent s'appliquer à tout ce qui entre dans le cadre de la Recommandation* (par exemple, aspects statique ou dynamique, utilisation ou fourniture d'un service, actions à effectuer, présence, absence ou forme d'un paramètre, etc.)

A.5 Un autre type de distinction est à faire entre les options relatives à l'utilisateur du service et celles relatives au fournisseur.

A.6 Dans un plus large contexte, le choix sera déterminé par des conditions qui sortent du cadre de la Recommandation* [par exemple, d'autres Recommandations* qui s'appliquent à la réalisation, les protocoles utilisés dans les couches ($N + 1$) et ($N - 1$), l'application prévue, les conditions de recette, le prix cible de la réalisation, etc.]. Toutefois ces conditions n'ont pas d'incidence sur la conformité à la Recommandation* dans laquelle figure l'option.

ANNEXE B

(à la Recommandation X.290, partie 2)

Directives à l'intention des rédacteurs de Recommandations* de protocole, visant à faciliter les tests de conformité

B.1 *Introduction*

La présente annexe donne des directives, essentiellement à l'intention des concepteurs de nouvelles Recommandations* de protocole, visant à faciliter les tests de conformité en garantissant une compréhension très claire des conditions de conformité.

B.2 *Directives relatives à l'objectif et au domaine d'application*

B.2.1 La précision du paragraphe Objectif et domaine d'application sert de règle générale en matière de précision pour le reste de la Recommandation*. Les spécifications stipulées dans la Recommandation* devraient être compatibles avec l'objectif et le domaine d'application et *vice versa*.

B.2.2 L'objectif doit distinguer clairement entre les trois types suivants d'informations fournies dans la Recommandation* de protocole:

- a) la définition des procédures de communication à suivre au moment de la communication;
- b) les conditions que doivent remplir les fournisseurs de réalisations pour la mise en œuvre des procédures;
- c) les directives concernant la façon de réaliser les procédures.

Les directives sur la façon de réaliser les procédures ne constituent pas des spécifications additionnelles et n'ont aucune incidence sur la conformité. Si de telles directives sont incluses, l'objectif devra le préciser et indiquer comment elles peuvent être distinguées des spécifications de la Recommandation*. Cette distinction est plus facile à faire si les directives sont séparées des spécifications. La méthode recommandée pour séparer les directives dans les documents de l'ISO est de les mettre en note ou en annexe.

B.2.3 Il doit être clairement indiqué à qui la Recommandation* s'adresse.

B.2.4 Le moment auquel la Recommandation* s'applique doit être clairement indiqué.

Les procédures de protocole s'appliquent entre deux partenaires en communication au moment de la communication. S'il y a un risque d'ambiguïté à propos des partenaires concernés, ce point doit être résolu dans l'objectif.

Il est préférable que les Recommandations* de protocole soient rédigées en sorte que les obligations incombent à un seul partenaire en communication (le «premier» partenaire en communication dans ce cas) au profit d'un ou plusieurs autres partenaires en communication (les «seconds» partenaires en communication). Quand deux partenaires en communication (ou plus) sont tous censés communiquer conformément à la Recommandation*, celle-ci est d'abord appliquée à un partenaire, considéré comme le «premier», puis à un ou plusieurs autres à tour de rôle. Cela garantit que, si les procédures ne sont pas respectées, le partenaire en faute est clairement identifié.

B.2.5 Si des directives sont formulées en ce qui concerne des facteurs qui ne sont pas définitivement normalisés, l'objectif doit préciser clairement que ces directives peuvent être ignorées sans que la conformité en soit affectée.

B.2.6 Les aspects qui sont exclus de l'objectif doivent être identifiés clairement.

Tous les facteurs concernant les procédures ou les produits qui les mettent en œuvre n'ont pas besoin d'être normalisés; il est certes souvent souhaitable de laisser une certaine liberté au réalisateur. Par exemple, il peut être souhaitable d'omettre, dans une Recommandation* de protocole, des spécifications de valeurs explicites de délais de temporisation et de les remplacer par des directives.

L'objectif doit préciser clairement quels aspects sont définitivement normalisés, ceux qui sont visés par des directives mais non par des obligations et ceux qui ne sont pas pris en considération par la Recommandation*. Tous les aspects donnant à penser qu'ils devraient être traités, du fait qu'ils se rapportent étroitement à des aspects qui sont normalisés, nécessitent une mention explicite.

B.2.7 Toutes les options devraient, si possible, être identifiées clairement dans l'objectif.

Les options sont une des plus grandes sources de difficultés, mais malheureusement elles font nécessairement partie des Recommandations* de protocole. Elles se situent entre ce qui est normalisé et ce qui ne l'est pas. Elles seront traitées de façon plus approfondie ci-après. A ce stade, ce qui est important est que les options ne soient pas enfouies profondément à l'intérieur de la Recommandation* mais énoncées clairement au début. Si le nombre et la nature détaillée des options rendent ceci impraticable, il faut se demander sérieusement si une telle complexité est réellement nécessaire. Des options détaillées ne peuvent-elles pas être regroupées d'une façon ou d'une autre (par exemple en classes) pour simplifier la Recommandation*?

B.2.8 L'objectif et le domaine d'application doivent être revus après considération de l'ensemble de la Recommandation*.

Souvent, il n'est pas possible de satisfaire certaines des directives ci-dessus avant d'avoir considéré la suite de la Recommandation*. Il est donc en général nécessaire de revenir à l'objectif, de vérifier qu'il est réellement en accord avec le contenu de la Recommandation*. On constate que des paragraphes complètement en dehors de l'objectif ont été introduits dans le texte.

B.3 *Directives concernant les références*

B.3.1 Les Recommandations* de protocole OSI doivent se référer au modèle de référence OSI, aux Recommandations* de service pertinentes et à toute Recommandation* pertinente concernant des conventions de protocole, des directives ou des techniques de description formelle.

B.3.2 Il convient de définir clairement si la conformité à la Recommandation* de protocole nécessite la conformité à une partie d'une autre Recommandation*.

B.3.3 Il convient de préciser clairement si chaque référence concerne une version particulière de la Recommandation* mentionnée ou à chacune de ses versions successives.

Normalement, la version la plus récente est requise, mais ceci peut être source de difficultés car des modifications apportées à la Recommandation* citée en référence peuvent affecter la conformité à la Recommandation* en cours de rédaction.

B.4 *Directives concernant les spécifications et les options*

B.4.1 Le statut de chaque spécification doit être non ambigu.

Comme les spécifications facultatives et conditionnelles sont très courantes, on a tendance à considérer tout ce qui peut être interprété comme facultatif comme l'étant effectivement.

B.4.2 Il devrait être possible à une instance de communication de se conformer à toutes les conditions de conformité dynamique obligatoires.

B.4.3 Les conditions dans lesquelles les spécifications s'appliquent doivent être énoncées clairement.

B.4.4 Il ne doit pas être impossible au réalisateur ou au fournisseur de savoir quelles sont ces conditions.

B.4.5 Il ne doit y avoir aucune possibilité de confusion entre ce qui est dynamiquement facultatif et ce qui l'est statiquement.

Des conditions de conformité statique obligatoires peuvent concerner des caractéristiques dont l'utilisation au moment de la communication est facultative. Réciproquement, un message dont l'utilisation est obligatoire dans un contexte donné au moment de la communication peut faire partie d'un mécanisme de protocole dont la prise en charge est statiquement facultative.

B.4.6 Si la Recommandation* propose un «choix» d'options et que des restrictions s'appliquent aux combinaisons autorisées de ces options, alors ces restrictions doivent être spécifiées clairement. Elles doivent inclure l'identification de toutes exclusions mutuelles et de toutes limites maximales ou minimales de la gamme autorisée d'options.

B.4.7 Si la Recommandation* ne donne aucune règle de sélection des options, l'objectif doit préciser clairement que seule la gamme complète des options et les options individuelles sont normalisées, mais non la sélection.

B.4.8 Les options légitimantes doivent être évitées. On appelle ainsi des options qui permettent de déclarer conformes à une même Recommandation* des versions différentes et incompatibles d'une même chose. Bien qu'elles n'empêchent pas par elles-mêmes une compréhension objective de la conformité, elles peuvent empêcher d'atteindre les objectifs de l'OSI.

B.4.9 Aucune option ne devrait autoriser le réalisateur à ignorer d'importantes spécifications de la Recommandation*. De telles options dévaluent la Recommandation* et la signification de la conformité à celle-ci.

B.4.10 Si la Recommandation* contient des interdictions, celles-ci devraient être suffisamment précises pour être significatives.

En effet, beaucoup de Recommandations* comportent des sections où il est indiqué «faire tout cela et rien d'autre». De telles interdictions peuvent être dénuées de signification, car tous les protocoles véhiculent certaines informations qui ne sont pas normalisées, les «données de l'utilisateur» et chaque produit normalisé a des attributs qui ne le sont pas, par exemple le poids. Il risque d'être difficile de départager ce que la Recommandation* ne peut pas interdire et ce que ses rédacteurs désirent interdire, si les interdictions ne sont pas formulées clairement.

B.5 *Directives relatives aux unités de données de protocole*

B.5.1 L'ensemble autorisé de types de PDU et de codages des paramètres doit être défini clairement.

B.5.2 La plage autorisée de valeurs doit être définie pour chaque paramètre.

B.5.3 Toutes les valeurs qui sont extérieures à la plage autorisée stipulée doivent être déclarées explicitement comme non valides.

Sinon, certains soutiendront que ces valeurs sont non définies, mais autorisées, alors que d'autres rétorqueront qu'elles sont non valides.

B.5.4 Il faut définir clairement si les types de PDU non définis sont ou non autorisés.

Il est plus sûr de déclarer comme non valides tous les types de PDU non définis.

B.5.5 Les valeurs critiques non définies doivent être mentionnées explicitement comme telles dans l'objectif.

B.5.6 Il doit exister une procédure définie à suivre par le premier partenaire en communication, chaque fois qu'il reçoit un type de PDU ou un paramètre non valide ou non défini.

B.5.7 Il doit être possible de détecter si les procédures définies ont été suivies dans de tels cas. Si non, ce doit être parce que cela n'a pas d'importance.

Parfois les procédures à suivre à la réception d'une PDU non valide sont intentionnellement identiques à celles concernant la réception d'une PDU valide dans les mêmes circonstances. Par exemple, la procédure peut être de ne rien faire tant qu'un type spécifique de PDU n'a pas été reçu, en ignorant tout le reste. Dans de tels cas, il importe probablement peu que l'erreur ait été apparemment non détectée. Dans certains autres cas, l'intention peut être

effectivement de répondre aux cas d'erreur par un certain traitement spécial, mais la procédure a été si mal choisie qu'il en résulte que ce traitement ne peut pas être distingué de l'action effectuée dans les cas où il n'y a pas d'erreur.

B.5.8 Si, dans le codage des PDU, certains champs sont déclarés réservés, il faut alors déclarer clairement quelles valeurs sont, le cas échéant, autorisées ou non autorisées dans ces champs.

B.5.9 Si des paramètres en relation peuvent être véhiculés dans des PDU séparées, alors l'ensemble des relations autorisées entre les valeurs de ces paramètres doit être précisément et clairement défini.

B.5.10 Si le codage des paramètres autorise leur spécification dans n'importe quel ordre et que le format des PDU impose des restrictions sur les ordres autorisés, ces restrictions devront être clairement formulées. Il devrait être admis que si de nombreux ordres différents sont autorisés, un échantillon important de ces différents ordres doit être testé. L'augmentation de complexité des tests de conformité qui en résultera devrait être compensée par certains avantages que permet cette liberté.

B.5.11 L'ordre dans lequel les bits, les octets, etc. pourraient être véhiculés dans le protocole sous-jacent doit être établi clairement.

Par exemple un entier de deux octets transportera-t-il en premier l'octet de plus fort poids ou de plus faible poids? Il est surprenant de voir combien souvent on peut laisser échapper d'aussi simples causes d'ambiguïté.

B.5.12 Les relations entre les SDU doivent être définies clairement.

B.6 *Directives relatives aux états*

B.6.1 Les procédures de protocole sont souvent définies par une méthode d'états finis, formalisée ou non. La spécification de ces états est souvent incomplète.

B.6.2 Chaque état doit être défini clairement.

B.6.3 Si des événements peuvent se présenter uniquement dans un sous-ensemble des états possibles, l'occurrence possible d'un événement doit être distinguée d'une occurrence valide.

B.6.4 Les actions et les transitions d'état requises doivent être définies pour chaque couple possible état/événement. En particulier, elles doivent être définies pour les couples état/événement possibles mais non valides.

B.7 *Directives relatives aux techniques de description formelle*

B.7.1 Les spécifications suivantes s'appliquent uniquement aux Recommandations* qui incluent une description formelle. Des Recommandations* précises et non ambiguës peuvent être rédigées sans l'aide de techniques de description formelle (FDT), mais elles sont vivement recommandées dans les Recommandations* complexes telles que les descriptions formelles de protocoles. Il faut toutefois réaliser qu'elles peuvent elles-mêmes être à l'origine de problèmes concernant la conformité.

B.7.2 Il convient d'indiquer clairement si les descriptions formelles constituent une partie essentielle de la Recommandation* ou sont fournies uniquement à titre de directives.

Il est très important d'avoir une compréhension claire de l'état de la description formelle. Dans l'idéal, il ne devrait pas exister de désaccord entre le texte et la description formelle, mais ceci étant très difficile à réaliser en pratique, il importe que le lecteur sache lequel a la priorité. Si une description formelle est fournie uniquement à titre indicatif, elle ne peut pas définir des conditions de conformité.

B.7.3 La FDT doit être bien définie, stable et comporter des références correctes.

B.7.4 Si la description formelle est utilisée pour la définition de certaines spécifications de la Recommandation*, mais non de toutes, il doit être indiqué clairement que le texte inclut des spécifications qui ne sont pas couvertes par la description formelle, et ces spécifications additionnelles doivent être identifiées clairement.

B.7.5 Si la description formelle définit des spécifications ainsi qu'une façon autorisée de réaliser certains aspects du protocole, mais qu'il est prévu de laisser le réalisateur libre de réaliser ces aspects d'une autre façon, ceci constitue une surdéfinition. Cette situation est beaucoup trop courante dans les descriptions formelles et crée des difficultés en ce qui concerne la conformité. Si la description formelle est une partie essentielle de la Recommandation*, elle doit être limitée par un texte qui indique où se trouvent de telles surdéfinitions et quelles sont les spécifications réelles.

Le problème se présente en général parce qu'une description formelle décrit le comportement interne d'une réalisation idéale et non le comportement externe observable requis. Seul le comportement externe observable peut être testé et c'est donc sur lui seul que doivent porter les conditions de conformité. Il serait préférable d'utiliser pour spécifier des conditions une FDT différente de celle utilisée pour les directives à l'intention des réalisateurs.

B.8 Directives diverses

B.8.1 Des informations qui peuvent sembler évidentes doivent néanmoins être énoncées.

Si une chose est omise parce qu'elle est «évidente», certains lecteurs supposeront qu'elle est requise du fait de son «évidence»; d'autres supposeront qu'elle est omise pour laisser le champ libre aux réalisateurs. Par exemple, l'existence d'un total de contrôle implique-t-elle qu'il doit être vérifié?

ANNEXE C

(à la Recommandation X.290, partie 2)

Conditions de conformité statique incomplètes

C.1 Historiquement, l'élaboration des Recommandations* de protocole a été entreprise parallèlement à la détermination de la signification de la conformité, et en particulier à des efforts visant à une meilleure compréhension de la distinction entre la conformité statique et dynamique.

C.2 Certaines des premières Recommandations* de protocole ne donneront donc pas une spécification complète des conditions de conformité statique. Un exemple typique de cet état incomplet est l'obligation de prendre en charge une fonction particulière, sans indiquer si ceci s'applique à l'expédition, à la réception ou à l'une et l'autre; ou l'absence de conditions précises de détection d'erreurs de protocole dans des messages entrants reçus.

C.3 En conséquence, ce qu'est une réalisation conforme peut donner lieu à des interprétations différentes.

C.4 Dans les Recommandations* futures, ou au moment de la révision des Recommandations* existantes, il sera nécessaire de fournir une spécification complète des conditions de conformité statique. Ceci devrait inclure la spécification de conditions applicables à des réalisations mettant en œuvre ou ne mettant pas en œuvre tout ce qui n'est ni toujours obligatoire, ni toujours optionnel.

C.5 A court terme, il est essentiel de modifier au moins les projets en cours pour clarifier la situation actuelle: il n'est pas acceptable que des Recommandations* soient diffusées sous une forme dans laquelle des spécifications de réalisation présentent une ambiguïté excessive. Il faut également envisager ce qui pourrait être fait quand les protocoles ont déjà atteint le statut de normes internationales ISO ou de Recommandations du CCITT.

C.6 Il n'est pas envisagé d'autre solution à court terme que d'accepter et de déclarer clairement que toutes les capacités non couvertes explicitement par les conditions de conformité statique sont facultatives et de réduire au minimum les risques de problèmes qui pourraient en résulter, en spécifiant que:

a) seules les réalisations qui:

- 1) mettent en œuvre tout ce qui est explicitement spécifié comme obligatoire; et
- 2) n'omettent rien en dehors de ce qui est explicitement déclaré comme optionnel, même s'il existe une clause générale du genre «si non spécifié, considérer comme facultatif»;

doivent être désignées comme «conformes» sans qualification;

b) toute réalisation qui:

- 1) met en œuvre tout ce qui est explicitement spécifié comme obligatoire; et
- 2) omet des éléments qui ne sont pas explicitement déclarés comme optionnels, peut-être à cause d'une clause générale du genre «si non spécifié, alors considérer comme facultatif»;

doit être décrite comme conforme à un sous-ensemble.

C.7 Les réalisations qui omettent quelque chose d'obligatoire ne sont pas conformes du tout.

Remarque – Un système conforme sans restriction ne sera pas nécessairement apte à l'interfonctionnement avec un autre système et ne fonctionnera pas nécessairement mieux qu'un système conforme à un sous-ensemble. Le système «parfait» peut refuser des PDU provenant d'autres systèmes qu'il considérerait comme incorrectes ou incomplètes. Il peut donc refuser ou couper des connexions.

Il convient donc d'accorder une attention particulière à la conformité en matière de détection d'erreurs de protocole, spécialement quand cette détection peut être explicitement ou implicitement facultative.

ANNEXE D

(à la Recommandation X.290, partie 2)

La notation combinée arborescente et tabulaire (notation TTCN)

D.0 Introduction

Dans la construction d'une suite de tests abstraite, une notation spécifique est utilisée pour décrire les tests élémentaires abstraits. Cette notation de tests peut être une notation informelle (sans sémantique définie précisément) ou une technique de description formelle (FDT). Cette annexe décrit une notation informelle appelée la notation combinée arborescente et tabulaire (TTCN).

TTCN répond aux objectifs suivants:

- a) fournir une référence commune pour l'évaluation d'autres notations de tests et un outil d'aide à l'analyse des problèmes de conception des tests élémentaires et des suites de tests;
- b) fournir une base pour la traduction des tests élémentaires dans d'autres notations de tests;
- c) être un langage permettant des descriptions de comportements appropriées à la spécification de tests élémentaires et de modules de tests.

Une suite de tests peut être envisagée comme une hiérarchie descendante, allant de la suite de tests complète aux événements de test (voir le § 8.1 de la partie 1). Avec TTCN, les types de base d'événements de test sont supposés être des primitives de service abstraites (ASP), des primitives locales abstraites (ALP) et des événements de temporisateurs.

Les tests élémentaires abstraits peuvent également être exprimés en termes de PDU (en utilisant un mécanisme d'abréviations décrit au § D.5.11).

D.1 Composants d'une suite de tests TTCN

Une suite de tests écrite en TTCN est composée des quatre sections suivants, dans l'ordre:

- a) Présentation (§ D.4)

Informations nécessaires à la présentation générale et à la compréhension de la suite de tests, comme les références du test et une description générale de son objet.

- b) Déclarations (§ D.5)

Description de l'alphabet des événements à utiliser dans la suite de tests (par exemple: des ASPS, des temporisateurs, des ASP, des PDU, des ALP et leurs paramètres). Cette partie contient les définitions de toutes les abréviations à utiliser dans la suite de tests.

- c) Partie dynamique (§ D.6)

Tableaux contenant des arbres de comportement exprimés principalement en termes d'occurrences d'ASP aux points de contrôle et d'observation. Un ensemble de descriptions des comportements principaux est suivi d'un ensemble de descriptions des comportements par défaut.

- d) Contraintes (§ D.8)

Cette partie spécifie les valeurs des ASP, des PDU et de leurs paramètres utilisés dans la partie dynamique.

D.2 Forme syntaxique de TTCN

TTCN est présentée sous une forme graphique (TTCN-GR) lisible par l'homme.

Remarque – Une forme lisible par la machine (TTCN-MP) adaptée à la transmission de descriptions TTCN entre machines et éventuellement à d'autres traitements automatiques est réservée pour étude ultérieure.

D.3 Conventions

D.3.1 Formulaires

La notation TTCN-GR est définie à l'aide de plusieurs types de tableaux. Les conventions suivantes seront utilisées pour la description de la structure de ces tableaux:

- a) les textes imprimés en gras (**cette police**) ou les textes dans cette police apparaîtront tels quels dans chaque tableau réel;

- b) les textes en italiques (*cette police*) n'apparaîtront pas tels quels. Les italiques indiquent les symboles devant être remplacés par un texte réel.

Des commentaires peuvent, en outre, figurer dans des formulaires hors des champs réservés spécialement aux commentaires: ils sont délimités par les symboles /* et */.

D.3.2 *Types de parenthèses*

Les termes utilisés pour désigner les divers types de parenthèses sont définis dans la figure D-2/X.290, partie 2.

Crochets: [. . .]
Accolades: { . . .}
Parenthèses: (. . .)

FIGURE D-2/X.290, Partie 2

Types de parenthèses

D.3.3 *Conventions de dénomination*

D.3.3.1 *Références des groupes de tests et des tests élémentaires*

- a) La référence (nom) d'un groupe de tests peut être de l'une des formes présentées à la figure D-3/X.290, partie 2 et illustrées par l'exemple D-2.

(Suite-Tests)/(Groupe-Tests)/
(Suite-Tests)/(Groupe₁-Tests) . . . / (Groupe_n-Tests)/

FIGURE D-3/X.290, Partie 2

Références de groupes de tests

Exemple D-2 – Référence d'un groupe de transport:

Transport/Classe0/Etab/Con/

- b) La référence (le nom) d'un test élémentaire est de l'une des formes présentées à la figure D-4/X.290, partie 2 et illustrées par l'exemple D-3.

(Suite-Tests)/(Tests-Elémentaires)/
(Référence-Groupe-Tests) . . . / (Test-Elémentaire)/

FIGURE D-4/X.290, Partie 2

Références de tests élémentaires

Exemple D-3 – Références de tests élémentaires de transport:

Init/Transport

Transport/Classe0/Etab/Con/Init-LT

- c) Les références des groupes de tests et les références des tests élémentaires définissent la structure de la suite de tests.

D.3.3.2 *Références de tests élémentaires*

- a) Les tests élémentaires sont attachés à un point donné de la structure de la suite de tests, telle que définie par les références des groupes de tests et des tests élémentaires (voir le § D.3.3.1).
- b) Des tests élémentaires peuvent être regroupés à leur point d'attache en bibliothèques à organisation hiérarchisée.
- c) Une référence de tests élémentaires est donc de la forme:

(Point-Attache):(Structure-Bibliothèque)

où:

- 1) *(Point-Attache)* est:

A) soit *(Suite-Tests)* ;

B) soit *(Référence-Groupe-Tests)* telle que définie au § D.3.3.1;

C) soit *(Référence-Test-Elémentaire)* telle que définie au § D.3.3.1;

- 2) et *(Structure-Bibliothèque)* est:

A) soit *(Test-Elémentaire)* ;

B) soit *(Composant₁)/. . .(Composant_n)/(Module-Test)*

Exemple D-4 – Références de tests élémentaires de transport:

Transport:Module-Test-A

Transport/Classe0/Etab-Con/:Module-Test-B

Transport/Classe0/Etab-Con/Init:LT:Module-Test-C

Transport:Bib-Commun/Préambules/Module-Test-C

Remarque 1 – Les deux points (:) de la référence de test élémentaire séparent la première partie qui se réfère à un point de la structure d'une suite de tests déjà définie, de la seconde partie qui définit la structure de la bibliothèque.

Remarque 2 – Les composants permettent le regroupement des tests élémentaires, selon n'importe quelle hiérarchie, en des bibliothèques qui n'ont aucune influence sur la structure proprement dite de la suite de tests.

D.4 *Présentation de la suite*

Cette section doit au moins inclure les informations suivantes:

- a) références aux normes de base concernées;
- b) une référence aux PICS et PIXIT avec une indication sur la façon de les utiliser;
- c) une indication de la ou des méthodes de test qui s'appliquent à la suite de tests;
- d) un index complet de la suite de tests, avec la référence, l'identificateur, le numéro de page et l'objet de chaque test élémentaire et de chaque module de test de la suite de tests. Les objets des tests seront classés selon la structure de la suite de tests.

D'autres informations pouvant aider à comprendre la suite de tests, ainsi que la façon dont elle a été dérivée, pourront également être incluses sous forme de commentaires.

Ces informations seront fournies dans le format présenté à la figure D-5/X.290, partie 2.

D.5 Déclarations

L'objet de la section Déclarations est de décrire l'ensemble des événements de tests, ainsi que tous les autres attributs à utiliser dans la suite de tests. Tous les objets utilisés dans la partie dynamique doivent être déclarés dans la partie déclarations. Il existe deux sortes d'événements de tests:

- a) les primitives de service abstraites (ASP) qui se présentent aux points de contrôle et d'observation (PCO) utilisés par le testeur (§ D.5.8);
- b) les événements de temporisation (§ D.5.10).

D'autres attributs sont également spécifiés:

- a) les types définis par l'utilisateur (§ D.5.1.3);
- b) les opérateurs définis par l'utilisateur (§ D.5.3);
- c) les paramètres de la suite de tests (§ D.5.4);
- d) les constantes globales (§ D.5.5);
- e) les variables globales (§ D.5.6);
- f) les PCO (§ D.5.7);
- g) les paramètres des ASP (§ D.5.8);
- h) les types de données (comprenant les PDU et leurs paramètres) (§ D.5.9);
- i) les abréviations (§ D.5.11).

Présentation de la suite			
Référence à des normes: ... Référence à la PICS: ... Référence aux PIXIT: ... Mode d'utilisation: ... Méthode(s) de test: ... Commentaires: ...			
Identificateur de test élémentaire	Référence de test élémentaire	Page	Objet
.	.	.	.
.	.	.	.
.	.	.	.
<i>(Identificateur de test élémentaire)</i>	<i>(Référence de test élémentaire)</i>	<i>(Page)</i>	<i>(Objet)</i>
.	.	.	.
.	.	.	.
.	.	.	.

FIGURE D-5/X.290, Partie 2

Formulaire de présentation d'une suite

D.5.1 Types généraux TTCN

TTCN utilise un certain nombre de types prédéfinis et de mécanismes, qui permettent la définition de types déclarés par l'utilisateur. Ces types peuvent être utilisés tout au long de la suite de tests et il peut y être fait référence quand sont définis des variables, des constantes, des paramètres d'ASP, des paramètres de PDU ou des paramètres de la suite de tests.

D.5.1.1 *Types prédéfinis*

Plusieurs types d'utilisation commune sont prédéfinis pour TTCN. Il peut y être fait référence même s'ils n'y apparaissent pas dans une déclaration de types d'une suite de tests. Tous les autres types utilisés dans une suite de tests doivent être nommés dans les déclarations des types définis par l'utilisateur (conformément au § D.5.1.2) et dénotés par leur nom.

- a) **Type prédéfini Entier:** type dont les valeurs distinguées sont les entiers positifs et négatifs, y compris zéro (comme valeur distinguée unique).
- b) **Type prédéfini Chaîne binaire:** type dont les valeurs distinguées sont des séquences ordonnées de zéro, un ou plusieurs bits.
- c) **Type prédéfini Chaîne d'octets:** type dont les valeurs distinguées sont des séquences ordonnées de zéro, un ou plusieurs octets, dont chacun est une séquence ordonnée de huit bits.
- d) **Types prédéfinis Chaîne de caractères:** types dont les valeurs distinguées sont zéro, un ou plusieurs caractères d'un certain jeu de caractères. Les types de chaînes de caractères indiqués à la figure D-6/X.290, partie 2 peuvent être utilisés. Ils sont définis au § 2 de la Recommandation X.208.

Chaîne-Numérique

Chaîne-Imprimable

Chaîne-Télétext (Chaîne-T61)

Chaîne-Vidéotex

Chaîne-Visible (Chaîne-ISO646)

Chaîne-IA5

Chaîne-Graphique

Chaîne-Générale

FIGURE D-6/X.290, Partie 2

Types prédéfinis de chaînes de caractères

- e) **Type prédéfini Identificateur d'extrémité de connexion (CEId):** type consistant en un nombre non limité de valeurs distinguées.

D.5.1.2 *Types définis par l'utilisateur*

Des types spécifiques à une suite de tests peuvent être introduits par l'utilisateur de TTCN. Pour définir un nouveau type, il faut fournir les informations suivantes:

- a) un nom du type;
- b) le type de base (le cas échéant);
- c) une définition du type, fournie de l'une des façons suivantes:
 - 1) en donnant une référence précise du ou des chapitres d'une norme qui définit le type;
 - 2) en donnant une référence de type ASN.1 (Recommandation X.208) de la forme:
(référence-module).(référence-type)
 - 3) en énumérant l'ensemble des valeurs distinguées nommées comprenant le type (voir la figure D-7/X.290, partie 2);

4) en spécifiant un sous-ensemble des valeurs distinguées d'un autre type.

Ce qui peut être fait de plusieurs façons:

- A) en spécifiant une sous-plage du type prédéfini *Entier* ;
- B) en spécifiant une sous-plage d'un type énuméré;
- C) en restreignant la longueur d'un type prédéfini *Chaîne binaire*, *Chaîne d'octets* ou *Chaîne de caractères*.

Ces informations seront fournies dans le format présenté à la figure D-8/X.290, partie 2.

Définitions de types par l'utilisateur			
Nom	Type de base	Définition	Commentaires
.	.	.	.
.	.	.	.
.	.	.	.
(Nom)	(Type de base)	(Définition)	(Commentaires)
.	.	.	.
.	.	.	.
.	.	.	.

FIGURE D-7/X.290, Partie 2

Formulaire de définitions de type par l'utilisateur

Définition de type par l'utilisateur			
Nom	Type de base	Définition	Commentaires
Classes-Transport	Aucun	(Classe 0, Classe 1, Classe 2, Classe 3, Classe 4)	Classes utilisables pour une connexion de la Couche Transport

FIGURE D-8/X.290, Partie 2

Exemple de déclaration de type par l'utilisateur

D.5.2 Notation des valeurs

Les valeurs des divers types seront notées comme suit:

D.5.2.1 Valeurs des types prédéfinis

Les valeurs des types prédéfinis seront notées comme suit:

- a) **Valeurs d'entiers:** les valeurs du type *entier* seront représentées par un ou plusieurs chiffres. Le premier chiffre ne sera un zéro que si la valeur est zéro.
- b) **Valeurs de chaînes binaires et de chaînes d'octets:** les valeurs des types *chaîne binaire* et *chaîne d'octets* peuvent être représentées de l'une des façons suivantes:

- 1) par une liste de bits.

Dans ce cas, la valeur est représentée par un nombre quelconque (éventuellement zéro) de zéros et de uns, précédés d'une ' unique et suivis du couple de caractères 'B';

Exemple D-5 – '01101100'B

- 2) par une liste de semi-octets.

Dans ce cas, la valeur consiste en un nombre quelconque (éventuellement zéro) des caractères:

0 1 2 3 4 5 6 7 8 9 A B C D E F

précédés d'une seule ' et suivis du couple de caractères 'H. Chaque caractère est utilisé pour représenter la valeur d'un demi-octet en représentation hexadécimale.

Exemple D-6 – 'AB0196'H

- c) **Valeurs de chaînes de caractères:** les valeurs des types *chaînes de caractères* sont représentées par un nombre quelconque (éventuellement zéro) de caractères du jeu désigné par le type de la chaîne de caractères, encadrés de ". Si le type chaîne de caractères inclut le caractère ", il sera représenté par un couple "" dans la représentation de toute valeur.
- d) **Valeurs d'identificateurs d'extrémité de connexion:** les valeurs du type *CEId* n'ont pas de notation.

Remarque – Aucune n'est nécessaire puisque la seule opération définie sur le type *CEId* est l'égalité (voir § D.5.8).

D.5.2.2 Valeurs de types définis par l'utilisateur

Les valeurs des types définis par l'utilisateur seront représentées comme suit:

- a) La notation des valeurs des types introduits par référence à un paragraphe ou à des paragraphes d'une Recommandation* doit spécifier, dans le commentaire associé à la définition du type, comment les valeurs distinguées de ce type doivent être notées dans la suite de tests.
- b) Une valeur d'un type ASN.1 cité est notée en ASN.1 ou de l'une des deux façons suivantes:

- 1) une référence de la forme:

(référence-module).(référence-valeur)

- 2) en spécifiant une valeur ASN.1 du type mentionné.

Remarque – La méthode modulaire ASN.1 comporte des extensions permettant la spécification partielle d'une valeur (§ D.8.3).

- c) Une valeur d'un type énuméré sera représentée par son nom.
- d) Une valeur d'un type obtenu en prenant un sous-ensemble d'un autre type aura la même notation que le type complet.

D.5.3 Opérateurs

Des opérateurs spécifiques à une suite de tests peuvent être introduits par l'utilisateur de TTCN. Pour définir une nouvelle opération, il faut fournir les informations suivantes:

- a) le nom de l'opération;
- b) la signature de l'opération, comprenant:
 - 1) une liste des types d'entrées;
 - 2) le nom de chaque composant d'entrée;
 - 3) le type du résultat;
- c) une description de l'opération.

Cette définition utilise le format présenté à la figure D-9/X.290, partie 2.

Définitions d'opération	
Nom de l'opération:	<i>(Nom de l'opération)</i>
Types d'entrées:	Liste de <i>(Nom de type)</i>
Composants:	Liste de <i>(Nom)</i>
Type de résultat:	<i>(Nom de type)</i>

Description:	<i>(Description)</i>
--------------	----------------------

FIGURE D-9/X.290, Partie 2
Formulaire de définition d'opération

Les définitions de deux opérations sur une chaîne sont données dans les figures D-10/X.290, partie 2 et D-11/X.290, partie 2.

Définitions d'opération	
Nom de l'opération:	<i>sous-chaîne</i>
Types d'entrées:	<i>chaîne x entier x entier</i>
Composants:	<i>source, début, longueur</i>
Type de résultat:	<i>chaîne</i>

Description:	<i>sous-chaîne (source, début, longueur) est la chaîne de longueur longueur commençant à l'indice début de source</i>
Par exemple:	<div style="text-align: right;"> <i>sous-chaîne ("abcde",3,2) = "cd"</i> <i>sous-chaîne ("abcde",4,9999) = "de"</i> </div>

FIGURE D-10/X.290, Partie 2
Définition de l'opération sous-chaîne

Définition d'opération	
Nom de l'opération:	<i>longueur</i>
Types d'entrées:	<i>chaîne</i>
Composants:	<i>source</i>
Type de résultat:	<i>entier</i>

Description:	<i>longueur (source) est la longueur de la chaîne source</i> Par exemple: <i>longueur ("abcde") = "5"</i>
--------------	--

FIGURE D-11/X.290, Partie 2

Définition de l'opération longueur

Remarque – Une opération peut être comparée à une fonction d'un langage de programmation ordinaire. Mais les arguments d'une opération ne sont modifiés par aucun appel de cette opération: il n'y a pas d'effets de bords.

D.5.4 *Paramètres d'une suite de tests*

L'objet de ce paragraphe est de déclarer les constantes dérivées de la PICS ou du PIXIT qui paramètrent globalement la suite de tests. Ces constantes sont appelées paramètres de la suite de tests.

Remarque – Dans la plupart des cas réels de tests, les paramètres d'une suite de tests seront fixés à une certaine valeur lors du traitement de la PICS ou du PIXIT.

Dans ce paragraphe, les informations suivantes sont fournies pour chaque paramètre de la suite de tests:

- a) son nom;
- b) son type.

Ces informations doivent être données dans le format présenté à la figure D-12/X.290, partie 2.

Paramètre de la suite de tests		
Nom	Type	Commentaires
.	.	.
.	.	.
.	.	.
<i>(Nom)</i>	<i>(Type)</i>	<i>(Commentaires)</i>
.	.	.
.	.	.
.	.	.

FIGURE D-12/X.290, Partie 2

Formulaire des paramètres d'une suite de tests

D.5.5 Constantes globales

L'objet de ce paragraphe est de déclarer un ensemble de noms de valeurs qui ne dérivent pas de la PICS ou des PIXIT et qui seront constants pendant toute la suite de tests.

Dans ce paragraphe, les informations suivantes doivent être fournies pour chaque constante globale:

- a) son nom;
- b) son type;
- c) sa valeur.

Ces informations doivent être fournies dans le format présenté à la figure D-13/X.290, partie 2.

D.5.6 Variables globales

Une suite de tests peut utiliser un ensemble de variables globales, concernant à la fois la partie dynamique et les contraintes. Normalement, ces variables seront utilisées pour des références aux valeurs des champs de contraintes de PDU ou d'ASP, ou composants, de la partie dynamique. Ces variables se présentent comme dans un langage de programmation classique (par exemple comme compteurs).

Toutes les variables globales à utiliser dans une suite de tests doivent être déclarées. Les informations suivantes seront fournies pour chaque déclaration de variables:

- a) son nom;
- b) son type.

Ces informations seront fournies dans le format présenté à la figure D-14/X.290, partie 2.

Constantes globales			
Nom	Type	Valeur	Commentaires
.	.	.	.
.	.	.	.
.	.	.	.
<i>(Nom)</i>	<i>(Type)</i>	<i>(Valeur)</i>	<i>(Commentaires)</i>
.	.	.	.
.	.	.	.
.	.	.	.

FIGURE D-13/X.290, Partie 2
Formulaire des constantes globales

Déclarations de variables globales		
Nom de la variable	Type	Commentaires
.	.	.
.	.	.
.	.	.
<i>(Nom de variable)</i>	<i>(Type)</i>	<i>(Commentaires)</i>
.	.	.
.	.	.
.	.	.

FIGURE D-14/X.290, Partie 2
Formulaire de déclarations de variables globales

Initialement, toutes les variables sont non liées.

Des variables peuvent devenir liées (ou être à nouveau liées) dans les circonstances suivantes:

- a) quand la variable apparaît dans la partie gauche d'une déclaration d'affectation (§ D.6.7.1);
- b) quand la variable non liée apparaît dans une expression booléenne (§ D.6.7.1);
- c) quand la variable apparaît dans une référence de contraintes (§ D.8).

D.5.7 Déclarations de PCO

Ce paragraphe donne la liste de l'ensemble des points de contrôle et d'observation (PCO) à utiliser dans une suite de tests en indiquant l'emplacement de ces PCO dans l'environnement de test.

Remarque – La méthode de test choisie détermine les PCO nécessaires pour définir la suite de tests.

Les informations suivantes seront fournies pour chaque PCO utilisé dans la suite de tests:

- a) son nom:
le nom sera utilisé dans la section relative au comportement pour spécifier où se produisent les événements;
- b) une explication concernant le type de testeur placé au PCO et le rôle joué par ce testeur.

Ces informations seront fournies dans le format présenté à la figure D-15/X.290, partie 2.

Déclarations de PCO	
Nom	Rôle
.	.
.	.
.	.
(Nom)	(Rôle)
.	.
.	.
.	.

FIGURE D-15/X.290, Partie 2
Formulaire de déclarations de PCO

Exemple D-7 – Un exemple de déclaration de PCO est présenté dans l'exemple D-8.

Exemple D-8 – Déclarations de PCO:

Déclarations de PCO	
Nom	Rôle
L	SAP au testeur inférieur/service (N-1) [le testeur inférieur est l'utilisateur du service (N-1)]
U	SAP au testeur supérieur/service (N) [le testeur supérieur est l'utilisateur du service (N)]

Les points de contrôle et d'observation coïncident en général avec des SAP, mais tout point approprié, auquel les événements de test peuvent être contrôlés et observés, peut convenir. Il est en outre possible de définir un PCO correspondant à un ensemble de SAP, à condition que tous les SAP que comprend ce PCO soient:

- au même endroit (c'est-à-dire dans le testeur inférieur ou dans le testeur supérieur);
- des SAP du même service.

Quand un PCO correspond à plusieurs SAP, l'adresse de l'entité appelante (au niveau de l'initiateur) ou l'adresse de l'entité appelée (au niveau du destinataire) est utilisée pour identifier chaque SAP.

Exemple D-9 – Un exemple typique de PCO correspondant à plusieurs SAP serait un testeur inférieur Internet utilisant un PCO représentant tous les *points de connexion au sous-réseau* pour envoyer plusieurs PDU Internet sur différents itinéraires. Une variante de cet exemple utiliserait plusieurs PCO.

Il est également possible d'envisager qu'un seul SAP corresponde à plusieurs PCO. Dans ce cas, il y aurait un PCO par connexion.

Remarque – Il est alors plus facile d'identifier la connexion correspondant à chaque événement de test.

A noter enfin qu'un PCO peut n'être associé à aucun SAP. Ce pourra, par exemple, être le cas quand une couche est composée de sous-couches. (Par exemple dans la couche application ou dans les couches de niveaux inférieurs, quand un point de connexion à un sous-réseau n'est pas un SAP.)

D.5.8 Déclarations des ASP

Cette section donne la liste des ASP qui peuvent se présenter aux PCO indiqués au § D.5.7.

Normalement, les informations à déclarer se trouvent dans la définition de service appropriée. Mais, leur déclaration explicite permet d'ajouter un commentaire spécifique au test et à une suite de tests particulière, ainsi que de répondre aux cas où il n'existe aucune définition de service OSI* explicite (par exemple X.25).

Les informations suivantes seront fournies pour chaque ASP:

a) Son nom:

Si une abréviation est utilisée, le nom complet doit suivre entre parenthèses (tel qu'il est donné dans une spécification de service appropriée).

b) Le ou les PCO auxquels elle peut se présenter:

Tous ces PCO doivent avoir été déclarés dans la section Déclarations des PCO de la suite de tests.

c) Un identificateur d'extrémité de connexion est-il utilisé ou non pour distinguer différentes occurrences de l'ASP?

Si l'utilisation d'un identificateur d'extrémité de connexion est déclarée [voir c) ci-dessus], un paramètre nommé CEID de type *CEID* est disponible sans autre déclaration.

A la réception d'une ASP qui utilise ce paramètre, la valeur de CEID sera initialisée pour l'extrémité de connexion où l'ASP a été reçue. Cette valeur est alors utilisable pour le test élémentaire.

Exemple D-10 – Utilisation d'un CEID:

PCO? AN-ASP [ceid = 3]

d) Une liste des paramètres associés à l'ASP.

Les informations suivantes seront fournies pour chaque paramètre:

1) son nom:

si une abréviation est utilisée, alors le nom complet doit suivre entre parenthèses (tel qu'il est donné dans une spécification de service appropriée);

2) son type.

Les déclarations d'ASP ont un seul niveau de paramètres. Mais le type de ces paramètres peut être de n'importe quelle complexité (par exemple un type ASN.1 complexe).

Ces informations seront fournies dans le format présenté à la figure D-16/X.290, partie 2.

Exemple D-11 – La figure D-17/X.290, partie 2 présente un exemple du service de transport (Recommandation X.214). Il pourrait faire partie de l'alphabet des ASP utilisées pour décrire le comportement d'un testeur supérieur abstrait dans une suite de tests DS de transport de classe 0. CDA, CGA et QOS sont des types définis par l'utilisateur (protocole de la Recommandation X.224).

Déclaration d'une ASP		
ASP: <i>(ASP)</i>	PCO: <i>(PCO) (liste)</i>	CEId: Utilisé ou non

Information de contrôle du service		
Nom du paramètre	Type	Commentaires
.	.	.
.	.	.
.	.	.
<i>(Nom du paramètre)</i>	<i>(Type)</i>	<i>(Commentaires)</i>
.	.	.
.	.	.
.	.	.

FIGURE D-16/X.290, Partie 2
Déclaration de primitive de service abstraite

Déclaration d'une ASP		
ASP: CONreq (T-CONNECT request)	PCO: TSAP	CEId: UTILISÉ

Information de contrôle du service		
Nom du paramètre	Type	Commentaires
CdA (adresse de l'entité appelée)	CDA	... du testeur supérieur
CgA (adresse de l'entité appelante)	CGA	... du testeur inférieur
QoS	Dépend de la réalisation	Doit garantir que la classe 0 est utilisée

FIGURE D-17/X.290, Partie 2
Exemple de déclaration de primitive de service abstraite

D.5.8.1 Déclarations des ALP

Les ALP sont déclarées en utilisant un pro forma similaire à celui des ASP. Les informations suivantes doivent être fournies pour chaque ALP:

- a) son nom;
- b) le ou les PCO où elle peut se présenter;
- c) une description fonctionnelle de cette ALP;
- d) une liste des paramètres associés à cette ALP.

Les informations suivantes doivent être fournies pour chaque paramètre:

- 1) son nom:
 - si une abréviation est utilisée, le nom complet suivra entre parenthèses (tel que donné dans une spécification de service appropriée);
- 2) son type.

Ces informations doivent être fournies dans le format présenté à la figure D-18/X.290, partie 2.

Déclaration d'ALP		
ALP: <i>(ALP)</i>	PCO: <i>(Liste de PCO)</i>	Description: <i>(Description fonctionnelle)</i>
Information de contrôle du service		
Nom du paramètre	Type	Commentaires
. . . <i>(Nom du paramètre)</i> <i>(Type)</i> <i>(Commentaires)</i> . . .

FIGURE D-18/X.290, Partie 2
Déclaration de primitive locale abstraite

D.5.9 Déclarations des types de données

L'objet de cette section est de déclarer les types de données qui sont utilisés dans la suite de tests. Ces types seront principalement utilisés dans les déclarations de paramètres d'ASP.

Les déclarations de types de données les plus courantes sont les déclarations de PDU. Les autres peuvent comprendre, le cas échéant, des déclarations de type ASN.1.

D.5.9.1 *Déclarations des PDU*

La déclaration des PDU est similaire à celle des ASP. Les informations suivantes doivent être fournies pour chaque PDU:

- a) son nom:
si une abréviation est utilisée, le nom complet suit entre parenthèses (tel que donné dans une spécification de protocole appropriée);
- b) une liste des paramètres ou, plus généralement, des champs associés à la PDU.

Remarque – Pour pouvoir décrire des tests portant sur le codage des PDU, il peut être nécessaire d'inclure des champs (comme, par exemple, des indicateurs de longueur) dans la description de la PDU, même s'ils ne sont pas considérés comme des paramètres de la PDU dans la spécification du protocole.

Les informations suivantes doivent être fournies pour chaque paramètre:

- 1) son nom:
si une abréviation est utilisée, le nom complet suivra entre parenthèses (tel que donné dans une spécification de protocole appropriée);
- 2) son type.

Ces informations doivent être fournies dans le format présenté à la figure D-19/X.290, partie 2.

Déclaration de types de données	
PDU: <i>(PDU)</i>	Commentaires: <i>[Commentaires généraux (par exemple, restrictions d'emploi)]</i>

Information de contrôle du protocole		
Nom du champ	Type	Commentaires
.	.	.
.	.	.
.	.	.
<i>(Nom du champ)</i>	<i>(Type)</i>	<i>(Commentaires)</i>
.	.	.
.	.	.
.	.	.

FIGURE D-19/X.290, Partie 2
Formulaire de déclaration des types de données

Quand une référence précise à une description de type ASN.1 d'une PDU est plus appropriée, elle peut être utilisée à la place de toutes les informations ci-dessus. Dans ce cas, les informations sont fournies dans le format présenté à la figure D-20/X.290, partie 2.

Déclarations de types de données	
Nom de la PDU	Définition du type ASN.1
.	.
.	.
.	.
(Nom de la PDU)	(Définition de type ASN.1)
.	.
.	.
.	.

FIGURE D-20/X.290, Partie 2
Formulaire de déclaration de type de données

Exemple D-12 – La figure D-21/X.290, partie 2 présente un exemple de définition ASN.1 de type de PDU de FTAM (ISO 8571).

Déclarations de type de données ASN.1	
Nom de PDU	Définition de type ASN.1
INIT-F	ISO 8571-PDU.FTAM

FIGURE D-21/X.290, Partie 2
Déclaration de type de données ASN.1

D.5.10 Temporisateurs

Une suite de tests peut utiliser plusieurs types de temporisateurs. Ces types sont utilisés pour distinguer le délai de temporisation. Un test élémentaire peut utiliser n'importe quel nombre d'occurrences du même type de temporisateur.

Les informations suivantes seront fournies pour chaque type de temporisateur:

- a) le nom du type de temporisateur;
- b) le délai de temporisation, qui peut être spécifié comme une valeur ou une plage de valeur.

Ces informations doivent être fournies dans le format présenté à la figure D-22/X.290, partie 2.

Déclarations de temporisateurs		
Nom du type de temporisateur	Délai	Commentaires
.	.	.
.	.	.
.	.	.
(Nom du type de temporisateur)	(Délai)	(Commentaires)
.	.	.
.	.	.
.	.	.

FIGURE D-22/X.290, Partie 2

Formulaire de déclarations de temporisateurs

Exemple D-13 – La figure D-23/X.290, partie 2 présente un exemple de déclaration de temporisation.

Déclaration de temporisateurs		
Nom du type de temporisateurs	Durée	Commentaires
Temporisateur-Réception	1..3	Définit un ensemble de temporisateurs à utiliser lors de la réception des données

FIGURE D-23/X.290, Partie 2

Exemple de déclaration de temporisateurs

D.5.11 *Abréviations*

Ce paragraphe définit toutes les abréviations utilisées dans le reste de la suite de tests. On utilise les abréviations comme des macros, en effectuant des opérations de simples substitutions textuelles. Elles peuvent être utilisées tout au long d'une suite de tests.

Une abréviation peut remplacer tout fragment de texte contenu dans une case de tableau. Le rédacteur d'un test élémentaire s'assurera que l'expansion qui en résulte suit la syntaxe TTCN.

Une définition d'abréviation fournira les informations suivantes:

- a) un identificateur d'abréviation, ou jeton;
- b) son expansion, à substituer à chaque occurrence de l'identificateur tout au long de la suite de tests.

Ces informations doivent être fournies dans le format présenté à la figure D-24/X.290, partie 2.

Abréviations		
Abréviation	Expansion	Commentaires
.	.	.
.	.	.
.	.	.
(Abréviation)	(Expansion)	(Commentaires)
.	.	.
.	.	.
.	.	.

FIGURE D-24/X.290, Partie 2

Formulaire d'abréviations

Exemple D-14 – La figure D-25/X.290, partie 2 présente un exemple de déclaration d'abréviation d'une suite de tests de transport (Recommandation X.224).

Remarque – L'opérateur est défini au § D.6.8.

Déclaration d'abréviation		
Abréviation	Expansion	Commentaires
CR	N-DATAind [NSDU ~ CR-TPDU]	CR représente toute indication N-DATA dont l'unité de donnée du service de réseau est le codage d'une unité de données du protocole de transport de demande de connexion

FIGURE D-25/X.290, Partie 2

Exemple de déclaration d'abréviation

D.6 *Partie dynamique*

La partie dynamique constitue le corps de la suite de tests: elle contient les descriptions de comportements du test élémentaire et/ou des modules de test.

D.6.1 *Description du comportement dynamique*

Les informations suivantes doivent être fournies pour la description du comportement de chaque test élémentaire, module de test ou ensemble de modules de test en relation:

- a) Une référence: la référence donne le nom du test élémentaire ou une description de comportements de test. Une référence de test élémentaire doit se conformer aux spécifications du § D.3.3.1. Une référence de module de test doit se conformer aux spécifications du § D.3.3.2.
- b) Un identificateur: une référence reflète la structure de la suite de tests et de l'objet du test élémentaire ou du module de test, et risque donc d'être assez longue. Il est parfois désirable d'avoir un nom court pour un test élémentaire ou un module de test. Tel est le rôle de l'identificateur: il peut être utilisé, de façon interchangeable, à la place d'une référence. L'identificateur de test est unique dans une suite de tests déterminée.
- c) Une déclaration de l'objet: déclaration informelle de l'objet du test élémentaire ou du (ou des) module(s) de test.
- d) Une référence de comportements par défaut: référence à une description éventuelle du comportement par défaut, qui s'applique à cette spécification de comportement (§ D.6.17).
- e) Une description du comportement: cette section décrit le comportement du ou des testeurs en termes d'événements de test (et de leurs paramètres) dans les trois notations décrites au § D.6.2.

Ces informations doivent être fournies dans le format présenté à la figure D-26/X.290, partie 2.

Comportement dynamique				
Référence: <i>(Référence de test élémentaire ou de module de test)</i>				
Identificateur: <i>(Identificateur)</i>				
Objet: <i>(Objet)</i>				
Référence de comportement par défaut: <i>(Référence de comportement par défaut)</i>				
Description du comportement	Etiquette	Référence des contraintes	Verdict	Commentaires
. . . . <i>(Description du comportement)</i> <i>(Etiquette)</i> <i>(Référence des contraintes)</i> <i>(Verdict)</i> <i>(Commentaires)</i> . . .
Conditions de synchronisation <i>(facultatives)</i>				
Commentaires supplémentaires <i>(facultatifs)</i>				

FIGURE D-26/X.290, Partie 2
Formulaire de déclaration de comportement dynamique

D.6.2 La notation arborescente

La notation arborescente est utilisée pour les descriptions de comportement d'une suite de tests. Les descriptions de comportement sont des énumérations de séquences observables possibles d'événements de test.

Exemple D-15 – Supposons que les séquences suivantes d'événements peuvent se produire durant un test dont l'objet est simplement l'établissement d'une connexion, l'échange de certaines données et la fermeture de cette connexion:

- a) CONreq, CONcnf, DATreq, DATind, DISreq
- b) CONreq, CONcnf, DATreq, DATind, DISind
- c) CONreq, CONcnf, DATreq, DISind
- d) CONreq, CONcnf, DISind
- e) CONreq, DISind

Cet enchaînement peut être modifié à tout moment par le fournisseur de service sous-jacent. La notation arborescente factorise simplement les séquences initiales communes de l'ensemble complet. Avec la notation arborescente, ceci s'écrirait:

```
↑
P  Sens du temps →
o  EXEMPLE-ARBRE
s  CONreq
s  CONcnf
i  DATreq
b  DATind
i  DISreq
l  DISind
i  DISind
t  DISind
é  DISind
s
↓
```

Les événements de même niveau d'indentation sont les événements possibles, pouvant se produire à un moment donné. Les événements possibles doivent être indiqués dans l'ordre où le testeur devra s'attendre à les voir.

D.6.3 Événements de test

Les noms des événements de test à lancer par un testeur sont préfixés d'un point d'exclamation (!). De même, ceux qui sont acceptables pour un testeur sont préfixés d'un point d'interrogation (?).

Exemple D-16 – Avec cette convention, l'exemple d'arbre serait représenté comme suit:

EXEMPLE-ARBRE

```
!CONreq
?CONcnf
!DATreq
?DATind
!DISreq
?DISind
?DISind
?DISind
?DISind
```

Ces noms (composés d'un ! ou d'un ? suivis d'un nom d'événement) doivent être préfixés d'un des noms de PCO figurant dans la liste des PCO de l'arbre où ils apparaissent - sauf si la suite de tests utilise un seul PCO, auquel cas le préfixe de PCO peut être omis. Le nom de PCO est utilisé pour indiquer le PCO auquel l'événement de test peut se produire.

Dans les cas où le comportement ne peut pas être spécifié en termes d'événements TTCN (par exemple du fait de la méthode de test utilisée) les descriptions de comportement seront fournies en langage normal.

Exemple D-17 – Module de test d'une suite de tests génériques:

ARBRE-PARTIEL

```
?N-DATind[Données-Utilisateur^DR]
?!N-DATreq[Données-Utilisateur^DC]
+EPILOGUE
```

EPILOGUE

```
/*fermer toutes les connexions de réseau ouvertes*/
```

D.6.3.1 *?OTHERWISE*

Le pseudo-événement prédéfini (*PCO*)*?OTHERWISE* peut être utilisé pour noter un événement d'ASP ou d'ALP que le testeur peut recevoir à ce PCO.

Exemple D-18 – Supposons que les événements A, B et C puissent être reçus à un PCO donné:

```
?A[X=1]
?B
?OTHERWISE
```

```
un certain comportement. . .
? TIMEOUT
```

est similaire à

```
?A
?B
?A

un certain comportement. . .
?B
```

```
un certain comportement. . .
?C
```

```
un certain comportement. . .
?TIMEOUT
```

A signaler les points suivants concernant l'utilisation de *?OTHERWISE*:

- a) est toujours étendu à tous les événements possibles, pouvant être reçus au PCO; en effet, si un événement se présente au même niveau, une expression booléenne ou une condition de synchronisation peut l'empêcher de se produire;
- b) si un arbre utilise plusieurs PCO, chaque PCO doit être déclaré dans l'instruction *?OTHERWISE*;
- c) *?OTHERWISE* ne doit pas être la dernière possibilité d'un ensemble de possibilités;
- d) *?OTHERWISE* peut être associé à une expression booléenne et/ou à une affectation.

Exemple D-19 – Illustration d'un *?OTHERWISE* qualifié:

ARBRE-PRINCIPAL (X)

```
?A                                succès
?B[X=2]                            échec1
?OTHERWISE[X=2]                    échec1
?C                                  échec1
?OTHERWISE                          succès
```

si l'arbre est lancé avec X=2, alors:

```
A => succès
B => échec1
```

Tout autre événement => échec₁

si l'arbre est lancé avec X 2, alors:

```
A => succès
C => échec1
```

Tout autre événement => succès

- e) ?OTHERWISE n'empêche pas l'utilisation de comportements par défaut. Les comportements par défaut doivent être rattachés à l'ensemble des possibilités et sont examinés dans l'ordre. Autrement dit, ?OTHERWISE indique certains comportements par défaut mais pas nécessairement tous.

Exemple D-20 – Utilisation de ?OTHERWISE dans l'arbre principal et par défaut.

ARBRE-PRINCIPAL

PCO1? A

PCO2? B

PCO1? OTHERWISE

ARBRE-DEFAUT

PCO2

?OTHERWISE échec₁

PCO1?C échec

 ?TIMEOUT succès

Le premier et le troisième événement de l'arbre par défaut ne sont pas invalidés par le ?OTHERWISE de l'arbre principal.

D.6.4 *Noms d'arbre*

Une description de comportements doit contenir au moins un arbre de comportements. Chaque arbre de comportements est préfixé d'un nom d'arbre, qui est unique dans une description de comportements (voir § D.6.5).

Remarque – Beaucoup d'exemples comportent des noms d'arbre en gras. Le but recherché est la clarté typographique, sans aucune autre signification.

D.6.4.1 *PCO formels*

Quand une suite de tests comprend la prescription des comportements de plusieurs PCO, le nom d'arbre doit être suivi d'une liste des noms, entre crochets, des PCO (formels) utilisés dans l'arbre. Quand une suite de tests prescrit le comportement d'un seul PCO, cette liste de PCO peut être omise.

Exemple D-21 – Un arbre comprenant les PCO L et U serait nommé: **NOM-ARBRE[L,U]**.

D.6.4.2 *Paramètres formels*

Si un module de test utilise des paramètres d'entrée, une liste des paramètres formels de l'arbre peut apparaître après l'éventuelle liste des PCO. Ils seront entre parenthèses. TTCN n'utilise pas de paramètres de sortie.

Exemple D-22 – **NOM-ARBRE[L,U](X,Y)**

D.6.5 *Rattachement d'arbres*

Des arbres peuvent être rattachés à d'autres arbres en remplaçant un nom d'événement par le nom de l'arbre à rattacher, préfixé par un +. Le nom de l'arbre rattaché sera de l'une des formes suivantes:

- <Référence-Arbre>;
- <Référence-Arbre>[<PCO-Réels>];
- <Référence-Arbre>(<Paramètres-Réels>);
- <Référence-Arbre>[<PCO-Réels>](<Paramètres-Réels>).

Quand un arbre est attaché à un autre, tous les éléments réels sont remplacés par des événements formels, par le biais d'un simple remplacement de texte. Ainsi un arbre rattaché est analogue à une macro.

D.6.5.1 *Domaine de rattachement*

Des descriptions de comportement peuvent contenir plusieurs arbres. Toutefois, seul le premier arbre de cette description est accessible de l'extérieur. Tous les arbres suivants sont considérés comme des modules de test locaux à la description de comportement, et ne sont donc pas accessibles de l'extérieur. Les tests élémentaires ne sont, bien sûr, pas rattachables.

Ainsi, la <Référence-Arbre> sera de l'une des formes suivantes:

- <Nom-Arbre>

Dans ce cas, <Nom-Arbre> sera le nom de l'un des arbres de la description de comportement courante.

- <Référence-Module-Test>/<Nom-Arbre>

Dans ce cas, <Nom-Arbre> sera le nom du premier arbre de la description de comportement du module de test cité par la référence.

Comme d'habitude, un <Identificateur-Module-Test> équivalent peut être mis à la place de <Référence-Module-Test>.

Exemple D-23 – Le couple suivant d'arbres:

ARBRE-PRINCIPAL [L,U] L!CONreq + SOUS-ARBRE[U]	SOUS-ARBRE [X] X?CONind
---	-----------------------------------

est équivalent à:

ARBRE-COMPOSE [L,U] L!CONreq U?CONind
--

Comme le rattachement d'arbres est en fait une méthode d'abréviation, les variables utilisées dans les arbres rattachés prennent la valeur qu'elles ont au point de rattachement de l'arbre, et peuvent ensuite être utilisées dans des expressions d'affectation ou des expressions booléennes de l'arbre rattaché.

Les arbres rattachés peuvent avoir des paramètres. Les paramètres réels remplacent les paramètres formels lors du rattachement de l'arbre.

Exemple D-24 – Le couple suivant d'arbres:

ARBRE-PRINCIPAL (M:=1) + Sous-Arbre(M,2) (M:=3)	SOUS-ARBRE (X,Y) (X:=Y)
---	-----------------------------------

est équivalent à

ARBRE-COMPOSE (M:=1) (M:=2) (M:=3)
--

D.6.6 *Étiquettes et GOTO*

Un ensemble d'événements possibles peut être étiqueté en plaçant une étiquette dans la colonne «étiquette», du premier des événements possibles.

Si un ensemble d'événements possibles est étiqueté, il est autorisé d'aller à («GOTO»), cet ensemble à partir de tout point de l'arbre. Ceci est réalisé en plaçant → ou le mot clé GOTO suivi du nom d'une étiquette qui est définie dans le même arbre, immédiatement après un événement. Si l'événement se produit, le test continue avec l'ensemble de possibilités désigné par l'étiquette.

Exemple D-25 – La figure D-27/X.290, partie 2 illustre l'utilisation de GOTO.

Les événements, et pas nécessairement le premier d'un ensemble d'événements possibles, peuvent également avoir une étiquette concernant la spécification des conditions de synchronisation (§ D.6.11). De telles étiquettes ne doivent pas être utilisées comme objets d'un GOTO.

Dans le cas où un premier événement doit être étiqueté pour un GOTO et relativement à des conditions de synchronisation, une étiquette commune sera utilisée.

D.6.7 *Expressions*

Les termes d'une expression peuvent être:

- des constantes (y compris des paramètres d'une suite de tests);
- des variables liées;
- des valeurs des paramètres de l'événement associé (le cas échéant) désignées par le nom de paramètre donné dans la déclaration de cet événement;
- des références à des valeurs de champs codés de PDU de paramètres d'ASP. Ces références ont la forme:

<Paramètre-ASP>.<Nom-Champ>

Comportement dynamique				
Référence: Exemple de GOTO Identificateur: GT Objet: Illustrer l'utilisation de «GOTO» Référence des comportements par défaut: –				
Description du comportement	Etiquette	Référence des contraintes	Verdict	Commentaires
/* Bouclage dans des données retournées en écho. Succès si une donnée est reçue en réponse à chaque donnée envoyée.*/ EchoUntilDis IDATAreq ?DATAind → encore ?DISind ?DISind	Encore		Echec Succès	Echo non reçu

FIGURE D-27/X.290, Partie 2
Exemple illustrant l'utilisation de GOTO

D.6.7.1 Clauses d'affectation

Tout événement peut être suivi d'une série d'affectations. L'effet de la clause d'affectation est de lier la variable globale à la valeur de l'expression si, et seulement si, l'événement se produit.

Les affectations sont effectuées dans l'ordre où elles apparaissent dans la clause. L'expression ne doit pas contenir de variables non liées.

D.6.7.2 Expressions booléennes

Un événement peut être qualifié en plaçant une expression booléenne entre crochets à sa suite. La qualification est considérée comme signifiant que la combinaison de l'événement se produisant et de l'expression booléenne est satisfaite.

Si une expression booléenne et une clause d'affectation sont associées au même événement, l'expression booléenne figurera en premier.

Les termes utilisés dans une expression booléenne peuvent être pris parmi les suivants:

- des constantes (y compris des paramètres de suite de tests);
- des variables:
 - si des variables non liées apparaissent dans une expression booléenne, l'événement se produit seulement s'il existe des variables non liées qui satisfont cette expression booléenne. Dans ces conditions, si l'événement se produit, toutes les variables non liées deviennent liées à toutes les valeurs cohérentes avec leur type, qui satisfont l'expression booléenne;
- les valeurs de paramètres de l'ASP associée (le cas échéant) désignées par le nom de paramètre donné dans la déclaration d'ASP;
- des références à un codage de PDU dans la partie contraintes (§ D.8). Ce codage peut être qualifié en écrivant:

<Référence-PDU>[<Expression-Booléenne>]

où les termes de l'expression booléenne peuvent être des références à des champs de la PDU citée;

- e) des références à des valeurs de champs codés de PDU des paramètres d'ASP de la partie contraintes. Ces références ont la forme:

<Paramètre-ASP>.<Nom-Champ>

D.6.7.3 Association de clauses d'affectation et d'expressions booléennes à des événements

L'association d'un événement à une affectation, à une expression booléenne ou aux deux à la fois, est autorisée. Toutefois, seules les combinaisons suivantes peuvent être utilisées:

- a) *<Événement>* L'événement n'est pas qualifié.
- b) *<Événement>(<Affectation>)* L'affectation est exécutée seulement si l'événement se produit.
- c) *<Événement>[<Expression-Booléenne>]* L'événement peut se produire seulement si l'expression booléenne est satisfaite.
- d) *<Événement>[<Expression-Booléenne>](<Affectation>)* L'affectation est exécutée si et seulement si les deux conditions suivantes sont satisfaites:
 - 1) l'événement se produit;
 - 2) l'expression booléenne est satisfaite.

Comme toutes les variables se présentant dans l'expression booléenne deviennent liées, elles peuvent être utilisées dans l'expression qui est une partie de l'affectation.

Les expressions booléennes peuvent être décomposées en un ensemble continu d'expressions booléennes. Cette décomposition est commode quand des abréviations sont utilisées en même temps (§ D.5.11). De même, des affectations peuvent être décomposées en un ensemble continu d'affectations.

Exemple D-26 – PCO? CR [X=1] [Y<3] est équivalent à PCO? CR [(X=1) AND (Y<3)].

D.6.7.4 Clauses d'affectation et expressions booléennes non associées à des événements

Il est permis d'utiliser des clauses d'affectation et des expressions booléennes seules, sans événement associé.

Seules les compositions suivantes sont autorisées:

- a) *(<Affectation>)* L'affectation est exécutée seulement si l'événement se produit.
- b) *[<Expression Booléenne>]* L'événement peut se produire si l'expression booléenne est satisfaite.
- c) *[<Expression Booléenne>](<Affectation>)* L'affectation est exécutée seulement si l'expression booléenne est satisfaite.

D.6.8 L'opérateur de codage/décodage

L'opérateur de codage/décodage permet la spécification du codage des champs de PDU. Sa syntaxe est définie au § D.6.8.2. L'opérateur ~ peut être lu comme «est le codage de».

Exemple D-27 – soit l'événement:

N-SAP? N-DATind[Données-Utilisateur ~ CR-TPDU] CR1

signifie que l'événement convient si et seulement si nous recevons au N-SAP une N-DATAind dont le champ Données-Utilisateur contient le codage d'une CR-TDPU conforme à la contrainte CR1 (voir le § D.6.13).

Supposons que F1 est un champ de la contrainte CR1. Si l'événement se produit, alors F1 peut être dénoté par Données-Utilisateur F1.

Remarque – Si le contenu de F1 est une variable libre (§ D.8.2) ou une variable globale (§ D.5.6), l'événement se produisant liera cette variable (voir le § D.6.7.2).

De même:

N-SAP! N-DATAreq[Données-Utilisateur ~ CC-TPDU] CC1

signifie que le testeur envoie une N-DATAreq dont le champ Données-Utilisateur contient le codage d'une CC-TPDU conforme à la contrainte CC1.

D.6.8.1 Utilisation de pseudonymes

TTCN fournit un mécanisme simple de conservation des valeurs de toutes les PDU. Le pseudonyme est une variable implicitement liée qui peut être utilisée comme abréviation ou pour distinguer deux champs (de PDU ou d'ASP) de même nom.

Exemple D-28 – utilisation de pseudonymes:

N-SAP? N-DATAind[Données-Utilisateur UD1 ~ CR-TPDU[Données-Utilisateur UD2 ~ TM-PDU]] CR1, TM1

Les pseudonymes UD1 et UD2 sont liés à leurs Données-Utilisateur respectives [ou, plus précisément, liés aux contraintes CR(CR1) et TM(TM1)]. Si l'on suppose que F1 est un champ de CR-TPDU(CR1) et F2 un champ de la contrainte TM(TM1): ces champs peuvent alors être désignés comme UD1.F1 et UD2.F2.

D.6.9 Arbres parallèles

Il est parfois pratique de pouvoir décrire un comportement sous forme de descriptions séparées de comportements se produisant en parallèle. Normalement, chacune de ces descriptions se rapporte à un comportement à un PCO différent.

Les arbres parallèles sont notés comme suit:

|| **Arbre₁** , . . . , **Arbre_n**

Les règles suivantes s'appliquent:

- a) l'indice n doit être supérieur à 1;
- b) l'arbre dont le comportement est divisé en arbres parallèles doit avoir plusieurs PCO, au moins aussi nombreux que les arbres parallèles;
- c) tous les PCO de l'arbre courant appartiennent à un arbre parallèle et un seul;
- d) aucun temporisateur ne doit être armé ni mis en attente;
- e) aucun autre événement ne doit figurer au même niveau d'événements possibles;
- f) seul le premier arbre (c'est-à-dire l'Arbre₁) peut rendre un verdict;
- g) quand le traitement du premier arbre (c'est-à-dire Arbre₁) est terminé:
 - 1) si Arbre₁ se termine sans verdict rendu, le test élémentaire se termine avec ce verdict;
 - 2) si Arbre₁ se termine avec un verdict (c'est-à-dire qu'une feuille d'Arbre₁ est atteinte), alors le comportement continue comme si Arbre₁ était attaché.
 - A) il n'est pas tenu compte des Arbre₂ . . . Arbre_n;
 - B) les comportements ultérieurs de l'arbre principal sont attachés aux feuilles d'Arbre₁;
 - 3) si une feuille d'Arbre_i (i > 1) est atteinte:
 - A) tout événement se produisant ultérieurement à un PCO d'Arbre_i sera une erreur;
 - B) autrement (g)2 s'applique (c'est-à-dire qu'Arbre₁ doit se terminer quelque part).

Exemple D-29 – comportement des testeurs inférieurs et supérieurs considérés comme des arbres parallèles:

```

ARBRE-PRINCIPAL [UT,LT]
+ connexion-couche-liaison(LT)
  UT! N-CONreq
    || LT-ARBRE(LT),UT-ARBRE(UT)
      + déconnexion-couche-liaison(LT)                                succès

ARBRE-UT [UT]
  UT? N-CONcnf                                                         L
  UT! N-DTAreq → L
  ?Otherwise → L
  UT? Otherwise → L

ARBRE-LT [LT]
  LT? N-DTAind [demande-appel]
  LT! N-DTAreq [accepte-appel]
  LT? N-DTAind [paquet-données]

ARBRE-DEFAULT
  UT? Otherwise
  + déconnexion-couche-liaison                                       échec1

```

D.6.10 *Gestion des temporisateurs*

Les temporisateurs utilisés dans une suite de tests sont supposés être toujours dans l'un des états suivants:

- a) désarmé,
- b) armé,
- c) en attente.

D.6.10.1 *Opérations relatives aux temporisateurs*

Un ensemble «d'opérations», pouvant être utilisées comme des affectations, sont utilisées pour modéliser la gestion des temporisateurs. Ces opérations peuvent s'appliquer à:

- a) un ensemble de temporisateurs.
On le spécifie en faisant suivre l'opération relative aux temporisateurs du seul nom du type de temporisateur;
- b) un temporisateur individuel d'un ensemble donné de temporisateurs de même type.
On le spécifie en faisant suivre l'opération relative au temporisateur par le nom et un identificateur de ce temporisateur. Un identificateur de temporisateur est un nom de variable.

Remarque – Il n'est pas nécessaire de déclarer explicitement les identificateurs de temporisateurs. Ils sont déclarés implicitement lors de leur définition comme paramètres de l'opération **Start**.

D.6.10.2 *Définition des opérations relatives aux temporisateurs*

Les opérations définies pour les temporisateurs sont:

- a) **Start** <type de temporisateur> [, <id de temporisateur> [, <valeur>]]

L'opération **Start** est utilisée pour indiquer qu'un temporisateur ou un ensemble de temporisateurs désarmés sont armés. (Si le paramètre identificateur de temporisateur est omis, ceci doit être interprété comme l'armement d'un temporisateur du type indiqué. Mais, il n'est alors pas possible de manipuler ce temporisateur séparément des autres temporisateurs de ce type qui peuvent être définis.)

Le paramètre de valeur optionnel est utilisé pour affecter un délai de temporisation (c'est-à-dire une durée) à un temporisateur. Cette valeur n'appartient pas nécessairement à la plage des valeurs définies pour le temporisateur. Il est ainsi possible de tester diverses situations concernant le délai de temporisation. Autrement, tout délai appartenant à la plage des valeurs spécifiées dans la partie Déclaration peut être utilisé.

- b) **Cancel** <type de temporisateur> [, <id de temporisateur>]

L'opération **Cancel** est utilisée pour indiquer qu'un temporisateur armé (ou en attente) est désarmé. (Si le paramètre identificateur de temporisateur est omis, tous les temporisateurs du type donné en attente seront désarmés.)

- c) **Suspens** <type de temporisateur> [, <id de temporisateur>]

L'opération **Suspend** est utilisée pour indiquer qu'un temporisateur armé passe en attente. (Si le paramètre identificateur de temporisateur est omis, tous les temporisateurs du type donné passeront en attente.)

- d) **Resume** <type de temporisateur> [, <id de temporisateur>]

L'opération **Resume** est utilisée pour indiquer qu'un temporisateur mis en attente est réactivé (c'est-à-dire que le décompte du délai est repris). (Si le paramètre identificateur de temporisateur est omis, tous les temporisateurs du type donné seront enclenchés.)

D.6.10.3 *Pseudo-événement de temporisation*

En plus des opérations relatives aux temporisateurs, deux pseudo-événements de temporisation sont définis:

- a) le pseudo-événement **Timeout** qui a la forme suivante:

?Timeout <type de temporisateur> [, <id de temporisateur>]

Ce pseudo-événement peut être utilisé dans un arbre de comportements pour détecter l'expiration d'un délai de temporisation spécifié. Le paramètre identificateur de temporisateur peut être omis:

- 1) s'il existe un temporisateur et un seul du type défini;
- 2) s'il existe plusieurs temporisateurs du type défini, mais il n'est pas nécessaire de les distinguer entre eux.

D.6.10.4 *Le pseudo-événement Elapse*

Une instruction **Elapse** a la forme suivante:

Elapse <type de temporisateur> [, <valeur>]

Une instruction **Elapse** impose une limite supérieure à la durée pendant laquelle un arbre peut rester à un niveau de retrait donné (c'est-à-dire dans un état donné). Il convient de signaler les points suivants concernant l'utilisation d'**Elapse**:

- a) aucune expression booléenne ni condition de synchronisation ne doit être associée à l'instruction Elapse;
- b) si plusieurs **Elapse** sont utilisées, seule celle qui spécifie le plus court délai est prise en considération;
- c) il est recommandé (mais pas obligatoire) que l'expression **Elapse** soit, le cas échéant, la dernière d'un ensemble de possibilités;
- d) la possibilité spécifiée par **Elapse** se produit seulement si, durant le délai imparti:
 - 1) aucune autre possibilité n'est programmée;
 - 2) aucune erreur ne se produit (c'est-à-dire un événement sans autres possibilités).

Remarque – Bien que Elapse utilise un type temporisateur pour faciliter l'attribution d'une valeur et d'unités, il n'implique l'armement d'aucun temporisateur accessible par l'utilisateur.

Exemple D-30 – utilisation d'**Elapse**:

?A
?B
ELAPSE

signifie que le testeur attendra que l'un des événements ?A ou ?B se produise, ou que le délai ELAPSE expire.

D.6.11 *Synchronisation*

L'instruction de synchronisation donne la possibilité de spécifier l'ordre relatif des événements dans différents arbres, qui peuvent appartenir ou non à la même description de comportement. L'instruction de synchronisation est une expression booléenne de l'une des deux formes:

- a) $(Référence-Arbre_1)/(Étiquette_1) < (Référence-Arbre_2)/(Étiquette_2)$ OR
- b) $(Référence-Arbre_1)/(Étiquette_1) > (Référence-Arbre_2)/(Étiquette_2)$

où les prédicats doivent être interprétés comme:

- a) l'événement étiqueté par $(Étiquette_1)$ dans le module de test $(Référence-Arbre_1)$ se produit **avant** l'événement étiqueté $(Étiquette_2)$ dans le module de test $(Référence-Arbre_2)$;
- b) l'événement étiqueté par $(Étiquette_1)$ dans le module de test $(Référence-Arbre_1)$ se produit **après** l'événement étiqueté $(Étiquette_2)$ dans le module de test $(Référence-Arbre_2)$

Remarque 1 – La synchronisation est spécifiée en termes d'événements se produisant dans les arbres différents, et non de choix dans le même arbre.

Remarque 2 – La synchronisation peut uniquement être utilisée entre des arbres parallèles, qui peuvent ou non faire partie de la même description de comportements.

Remarque 3 – Une autre possibilité d'arbres parallèles synchronisés serait un arbre unique comportant plusieurs PCO.

D.6.11.1 *Interprétation des déclarations de synchronisation*

Une déclaration de synchronisation sera interprétée comme suit:

SI

- a) un événement d'un ensemble d'événements possibles a une étiquette L, ET
- b) qu'il existe une instruction de synchronisation concernant $(Référence-Arbre/Étiquette)$ dans sa partie droite ou gauche, ET
- c) que $(Référence-Arbre)$ désigne la description de comportement où se produit l'événement,

ALORS

Ajouter ce qui suit à l'expression booléenne utilisée pour décider si l'événement est sélectionné:

«ET l'événement désigné par l'étiquette de l'autre membre de l'instruction de synchronisation s'est déjà produit (respectivement ne s'est pas déjà produit).»

Une instruction de synchronisation ne signifie donc pas que l'arbre est «mis en attente» jusqu'à ce que la condition de synchronisation soit satisfaite, mais que l'événement concerné par la synchronisation n'est pas candidat pour une sélection à opérer à partir de l'ensemble courant de possibilités, si la synchronisation n'est pas satisfaite. Les règles normales de sélection d'un événement s'appliquent, en considérant la synchronisation comme une expression booléenne supplémentaire. Si aucun événement ne peut être sélectionné, une erreur peut alors se produire si un événement se présente au PCO qui ne peut pas être soumis au contrôle de flux. Le concepteur du test doit donc inclure l'événement approprié au même niveau de retrait.

Exemple D-31:

PREMIER-ARBRE		
LT!L-DTAreq [paquet DT avec bit D positionné]	L1	
LT?L-DTAind [P(R) pour ce paquet]	L2	échec ₁
LT?L-DTAind [P(R) < à ce paquet] → L2		
LT?L-DTAind [P(R) < à ce paquet]	L3	succès
SECOND-ARBRE		
UT?N-DTAind	L	

avec les conditions de synchronisation:

PREMIER-ARBRE/L3 > SECOND-ARBRE/L
 PREMIER-ARBRE/L2 < SECOND-ARBRE/L

D.6.12 La structure REPEAT

Ce paragraphe décrit un mécanisme des descriptions TTCN de comportement, permettant d'itérer un module de test un nombre variable de fois. La forme de la structure est:

REPEAT (*Référence-Arbre*) [(*Id-Variable*)]

et doit être suivie par un ensemble d'expressions booléennes (limites) possibles. A noter que:

- a) les limites n'ont pas besoin d'être mutuellement exclusives. La répétition se termine dès qu'une limite est atteinte;
- b) aucune autre événement ou pseudo-événement ne peut apparaître au même niveau de retrait que les limites;
- c) l'identificateur de variable Id-Var (quand il est spécifié)
 - 1) peut être utilisé dans l'arbre qui est répété;
 - 2) peut être utilisé à l'intérieur des limites;
 - 3) n'est pas une variable globale (c'est-à-dire que sa portée est locale par rapport à REPEAT);
 - 4) est implicitement de type entier.

D.6.12.1 REPEAT sans identificateur de variable

Si l'identificateur des variables n'est pas spécifié, alors la structure:

```

REPEAT Référence-Arbre (a1, . . . , an)
  [b1]
  . . .
  .
  .
  .
  [bn]
  . . .
    
```

sera interprétée comme signifiant:

+ Référence-Arbre (a_1, \dots, a_n)L
 [b₁]
 ...
 .
 .
 .
 [b_n]
 ...
 [vrai] → L

où (a_1, \dots, a_n) sont (le cas échéant) les vrais paramètres de l'arbre.

D.6.12.2 REPEAT avec indicateur de variable

Quand Id-Var EST spécifié, la structure:

REPEAT Référence-Arbre (a_1, \dots, a_n) [Id-Var]

se développe en:

+ arbre-repeat ($a_1, \dots, a_n, 0$)

et cet arbre-repeat est défini comme:

Arbre-repeat-fictif ($f_1, \dots, f_n, \text{Id-Var}$)
 + Référence-arbre ($f_1, \dots, f_n, \text{Id-Var}$) L
 (Id-Var:=Id-Var+1)
 [b₁]
 ...
 .
 .
 .
 [b_n]
 ...
 [vrai] → L

où (a_1, \dots, a_n) sont les paramètres réels de l'arbre (le cas échéant), (f_1, \dots, f_n) sont les paramètres formels de l'arbre et Id-Arbre (f_1, \dots, f_n) est redéfini comme Id-Arbre ($f_1, \dots, f_n, \text{Id-Var}$).

D.6.13 Référence des contraintes

Cette colonne permet de faire référence à une ASP, une ALP ou une PDU qui est définie dans la partie Contrainte du § D.8 – qui donne également d'autres informations sur les contraintes de codage.

Exemple D-32 – une référence de contrainte:

N-SAP? N-DATAind [Données-Utilisateur ~ CR-TPDU] N-DATAind(D1), CR-TPDU(CR1)

où N-DATAind et CR-TPDU sont définis dans les sections de déclaration de types d'ASP et de données de la suite de tests. Les contraintes D1 et CR1 sont définies dans la partie Contraintes de la suite de tests.

Si un événement est qualifié par une expression booléenne, et comporte également une référence de contraintes, ceci doit être interprété comme *l'événement se produit si et seulement si l'expression booléenne et la contrainte sont satisfaites*.

Si un événement est suivi d'une clause d'affectation et comporte également une référence de contraintes, une expression booléenne ou les deux, ceci doit être interprété comme *l'affectation est effectuée si et seulement si l'événement se produit*, conformément aux définitions données ci-dessus.

D.6.14 Verdicts

Un test élémentaire doit se terminer par un verdict. Ce verdict est rendu par rapport à l'objet du test et peut être l'un des suivants:

- a) succès
- b) échec₁
- c) échec₂ (objet du test réalisé, mais échec subséquent)

- d) échec₃ (erreur de protocole, mais objet du test non concluant)
- e) non concluant.

Un verdict peut être rendu sur un événement de test en utilisant la colonne verdict. Les entrées de cette colonne doivent contenir une des valeurs indiquées au paragraphe suivant ou être laissées vierges (c'est-à-dire aucun verdict). Ce cas particulier sera mentionné par la suite comme la valeur spéciale *aucun*.

D.6.14.1 Règles d'énonciation des verdicts

Les règles suivantes s'appliquent à l'énonciation des verdicts:

- a) à tout moment de l'exécution d'un test élémentaire, il existe un «verdict par défaut»;

Remarque – A ne pas confondre avec un verdict qui peut apparaître dans une description de comportement par défaut.
- b) au début d'un test, le verdict par défaut est *aucun*;
- c) quand un arbre est rattaché, son verdict par défaut se transforme comme indiqué dans le tableau de la figure D-28/X.290, partie 2. Dans cette figure, «courant» est le verdict par défaut courant et «nouveau» est le verdict rendu au point d'attache;
- d) quand le verdict rendu sur un événement (autre qu'un rattachement) n'est pas «aucun» (c'est-à-dire que la colonne verdict n'est pas vierge), le test élémentaire se conclut avec le verdict qui résulte de l'application de la table de préséance de la figure D-28/X.290, partie 2 au verdict par défaut et au verdict rendu sur cet événement;
- e) quand la feuille d'un arbre qui n'est pas l'arbre principal est atteinte, le test élémentaire se conclut par le verdict par défaut.

Remarque – Dans ce cas, aucun verdict explicite n'est rendu sur le dernier événement de l'arbre [autrement le § D.6.14.1 d) s'appliquerait]. C'est un cas d'erreur.

Exemple D-33 – utilisation d'un verdict attaché à un sous-arbre:

ARBRE-A

```
!CONreq
?CONcnf
  + Transfert-Données («Hello»)
    [ok = «oui»
      + au-revoir Succès
    [ok = «non»
      + au-revoir Echec
?DISind Non-concl.
```

	Nouveau					
Courant	aucun	Succès	Echec ₁	Echec ₂	Echec ₃	Non-concl.
aucun	aucun	Succès	Echec ₁	Echec ₂	Echec ₃	Non-concl.
Succès	Succès	Succès	Echec ₂	–	–	–
Echec ₁	Echec ₁	–	Echec ₁	–	–	–
Echec ₂	Echec ₂	–	Echec ₂	Echec ₂	–	–
Echec ₃	Echec ₃	–	Echec ₁	–	Echec ₃	–
Non-concl.	Non-concl.	–	Echec ₃	–	–	Non-concl.

FIGURE D-28/X.290, partie 2
Préséance des verdicts par défaut

D.6.15 *Commentaires*

Cette colonne contient des brèves remarques, ou des références aux commentaires étendus donnés au bas du tableau.

D.6.16 *Référence des comportements par défaut*

Pour mettre en évidence le chemin principal du déroulement d'un test, il est possible de spécifier, séparément de la description de comportements principale, les comportements par défaut résultant de tout événement qu'un testeur peut être appelé à traiter. La description de comportements principale de la partie dynamique est complétée en ajoutant à chaque ensemble de possibilités des descriptions de comportements résultants, d'après les descriptions de comportements par défaut.

Pour être complète, une spécification TTCN de test élémentaire doit spécifier le comportement résultant de chaque événement possible, dans la description de comportement principale ou au moyen d'une description associée de comportement par défaut.

Remarque – Ceci peut être réalisé simplement à l'aide de ?OTHERWISE.

Les comportements par défaut s'appliquent uniquement quand l'événement de test concerné est possible.

Exemple D-34 – le test élémentaire suivant pourrait être divisé en deux descriptions de comportements:

ARBRE-EXEMPLE

```
!CONreq
  ?CONcnf
    !DATreq
      ?DATind
        !DISreq
```

avec le comportement par défaut trivial (sans comportement résultant associé):

ARBRE-DÉFAUT

```
?DISind
```

La référence des comportements par défaut est utilisée pour associer un ensemble de comportements par défaut à la description de comportements principale.

D.6.17 *Spécification des comportements par défaut*

Les descriptions de comportements par défaut doivent être présentées en tableaux dans la section Comportements par défaut de la partie dynamique d'une suite de tests. Ces tableaux doivent être du format présenté à la figure D-29/X.290, partie 2.

Comportement dynamique par défaut				
Référence: <Référence de test élémentaire ou de module de test> Identificateur: <Identificateur> Objet: <Objet> Référence par défaut: <Référence par défaut>				
Description du comportement	Etiquette	Référence des contraintes	Verdict	Commentaire
<Description du comportement>	<Etiquette>	<Référence des contraintes>	<Verdict>	<Commentaire>
Commentaires étendus (<i>facultatif</i>)				

FIGURE D-29/X.290, partie 2
Formulaire de déclaration de comportement dynamique

Ces tableaux sont tout à fait analogues à ceux de la description de comportements principale, sauf les différences suivantes:

- l'arbre des comportements par défaut n'a pas de nom;
- il contient un arbre unique sans nom. (Les PCO et les paramètres ne peuvent pas être communiqués à un arbre de comportements par défaut.);
- il ne peut pas comporter d'entrée Conditions de synchronisation.

Remarque – Les références aux comportements dynamiques par défaut suivent les mêmes règles que les modules de test (§ D.3.3.2).

Exemple D-35 – les figures D-30/X.290, D-31/X.290 et D-32/X.290, partie 2, présentent une version légèrement augmentée de l'ARBRE-EXEMPLE. La combinaison des comportements par défaut présentés aux figures D-31/X.290 et D-32/X.290, partie 2, est exactement équivalente au comportement présenté à la figure D-33/X.290, partie 2.

Comportement dynamique				
Référence: Exemples/Grand-Arbre-Exemple Identificateur: GAE Objet: Illustration de l'utilisation des comportements par défaut Référence des comportements par défaut: Exemples/Défaut/Standard				
Description du comportement	Etiquette	Référence des contraintes	Verdict	Commentaires
GRAND-EXEMPLE-ARBRE !CONreq ?CONcnf IDATreq !DATind !DISreq			Succès	Demande Confirmation Emission données Accept données Déconnexion

FIGURE D-30/X.290, partie 2
Exemples de comportements dynamiques (partie 1 de 3)

Comportement dynamique par défaut				
Référence: Exemples/Défauts/Standard Identificateur: Std Objet: Premier niveau de comportements par défaut Référence des comportements par défaut: Exemples/Défaut/Standard				
Description du comportement	Etiquette	Référence des contraintes	Verdict	Commentaires
?RSTind !DISreq			Non-concluant	Réinitialisation Terminaison

FIGURE D-31/X.290, partie 2
Exemple de comportement par défaut (partie 2 de 3)

Comportement dynamique par défaut				
Référence: Exemples/Défauts/Défaut Identificateur: Def Objet: Second niveau des comportements par défaut Référence de comportements par défaut: Aucune				
Description du comportement	Etiquette	Référence des contraintes	Verdict	Commentaire
?DISind			Non-concluant	Prématuré

FIGURE D-32/X.290, partie 2
Exemple de comportement par défaut (partie 3 de 3)

Comportement dynamique par défaut				
Référence: Exemples/Défauts/Défaut Identificateur: Std Objet: Comportements par défaut combinés Référence de comportements par défaut: Aucune				
Description du comportement	Etiquette	Référence des contraintes	Verdict	Commentaires
?RETind !DISreq ?DISind ?DISind			Non-concluant Non-concluant Non-concluant	Réinitialisation Terminaison Terminaison Prématuré

FIGURE D-33/X.290, partie 2
Exemple de comportement par défaut (parties 2 et 3 combinées)

D.7 *Interprétation des arbres*

Ce paragraphe décrit l'interprétation des descriptions de comportements TTCN et l'attribution d'un verdict à un test élémentaire. Il comprend deux parties destinées à des catégories de lecteurs différents:

- la première partie donne un ensemble de règles, ainsi qu'un algorithme simple destinés à aider la compréhension de TTCN;
- la seconde partie donne un algorithme plus complet (les règles ont été intégrées à l'algorithme) destiné à aider la mise en œuvre de TTCN.

D.7.1 *Interprétation de TTCN (algorithme partiel)*

- a) un arbre TTCN comporte un nombre de niveaux hiérarchiques d'événements supérieur ou égal à un. L'exécution des événements d'un certain niveau de l'arbre se termine par:
 - 1) SOIT l'énoncé d'un verdict;
 - 2) SOIT l'occurrence d'une erreur de test élémentaire (c'est-à-dire qu'une feuille est atteinte sans qu'aucun verdict ait été rendu, ou un événement imprévu est reçu à un PCO);
- b) chaque niveau comporte un nombre d'événements possibles supérieur ou égal à un;
- c) nous supposons ici que:
 - 1) l'expansion de l'arbre a déjà été réalisée conformément aux règles générales données au § D.6.5;
 - 2) l'ensemble des comportements par défaut à appliquer a été déterminé conformément aux règles données au § D.6.14.1;
 - 3) les règles des verdicts par défaut dans le cas d'un sous-arbre sont données dans le § D.6.14.1;
- d) l'algorithme de détermination de l'action à effectuer à un niveau donné est le suivant:
 - 1) ajouter les comportements par défaut à l'ensemble des possibilités;
 - 2) utiliser la procédure du § D.7.3 pour déterminer s'il y a identification;
 - 3) si la procédure du § D.7.3 retourne *aucune identification*, alors TERMINER (erreur);
 - 4) si une correspondance est trouvée, alors:
 - A) si un verdict a été rendu pour l'événement qui cause la correspondance, ou si cet événement est la feuille d'un sous-arbre à laquelle est associé un verdict par défaut, terminer alors le test élémentaire, et retourner le verdict final, rendu conformément au tableau présenté à la figure D-29/X.290, partie 2;
 - B) autrement, prendre comme niveau de retrait suivant, celui qui suit l'événement qui a causé la correspondance et:
 - i) si ce niveau désigne un ensemble de possibilités non vides, relancer le traitement de ce niveau;
 - ii) autrement, TERMINER (erreur).

D.7.2 *Règles relatives aux comportements par défaut*

- a) les comportements par défaut d'un sous-arbre ont la préséance sur (c'est-à-dire doivent être envisagés avant) les comportements par défaut de l'arbre auquel est rattaché le sous-arbre;

Exemple D-36 – comportements par défaut dans un arbre principal et un arbre rattaché:

ARBRE-PRINCIPAL

?A

+ Module-1

ARBRE-DÉFAUT

?B

échec₁

?E

échec₁

Module-1

?C

| ?D

Défaut-Module-1

?B

succès

et les séquences:

?A ?C ?B => succès

?A ?C ?B => échec₁

- b) les comportements par défaut d'un sous-arbre ne sont pris en considération qu'au moment de l'accès effectif à ce sous-arbre (c'est-à-dire à un niveau de retrait supérieur à ceux des événements qui ont provoqué l'entrée dans ce sous-arbre);
- c) les comportements par défaut cessent de s'appliquer à la fin d'un arbre et ne doivent pas être ajoutés à l'ensemble de possibilités vides qui suit une feuille de l'arbre. Ceci est vrai pour l'arbre principal comme pour le sous-arbre;

Exemple D-37:

ARBRE-PRINCIPAL

?A
 ?B
 ?C échec₁
?OTHERWISE succès

ARBRE-DÉFAUT

?OTHERWISE non-concluant

et la séquence ?A ?B ?ANYTHING est non valide (c'est-à-dire que le test élémentaire est incomplet);

- d) quand des comportements par défaut sont utilisés dans un sous-arbre et que se produit un événement conforme à ce comportement, si la séquence spécifiée par le comportement par défaut n'a pas de verdict, l'exécution continuera dans l'arbre principal quand une feuille de l'arbre par défaut aura été atteinte, sauf, bien sûr, si le sous-arbre comporte un verdict par défaut!

Exemple D-38 – arbre par défaut avec verdicts:

ARBRE-PRINCIPAL

?A
 + Module-1
 ?X succès
?B échec₁

Module-1

?D
 ?E

Défaut

?B
 ?C

et la séquence ?A ?D ?B ?C ?X => succès

D.7.3 Identification des événements

Les règles permettant d'identifier si un événement, parmi un ensemble d'événements possibles, convient (c'est-à-dire est évalué comme s'étant produit) sont indiquées ci-après pour chaque type possible d'événement ou de pseudo-événement TTCN:

- a) **!(ASPid | ALPid | PDUid)** – pour qu'un événement d'ÉMISSION convienne, il doit être possible d'envoyer l'ASP, l'ALP ou la PDU spécifiée (par exemple, en utilisant le contrôle de flux). Si un PCO a été spécifié, cette détermination est faite en considérant ce PCO. En outre, toutes les expressions booléennes et/ou contraintes qualificatives s'appliquant à l'ASP, l'ALP ou à la PDU doivent être satisfaites;
- b) **?(ASPid | ALPid | PDUid)** – pour qu'un événement de RÉCEPTION convienne, l'ASP, l'ALP ou la PDU spécifiée doit avoir été reçue et toutes les contraintes spécifiées doivent être satisfaites. Si un PCO a été spécifié, l'ASP, l'ALP ou la PDU doivent avoir été reçues à ce PCO. En outre, toutes les expressions booléennes et/ou contraintes qualificatives s'appliquant à l'ASP, l'ALP ou la PDU doivent être satisfaites;
- c) **?TIMEOUT** – le pseudo-événement TIMEOUT convient si le délai d'un temporisateur de type nommé a expiré. Si un identificateur optionnel de temporisateur est spécifié, seule l'expiration du délai de ce temporisateur est prise en considération;
- d) **ELAPSE** – pour que l'événement ELAPSE convienne, le délai du temporisateur qui a été implicitement armé (par la spécification d'ELAPSE) doit avoir expiré avant qu'un des événements antérieurs à ELAPSE n'ait été identifié;
- e) **?OTHERWISE** – pour que l'événement OTHERWISE convienne, un événement qui ne corresponde à aucune des possibilités antérieures à OTHERWISE doit avoir été reçu. Si un PCO est spécifié, l'événement doit avoir été reçu à ce PCO;
- f) **ATTACHE** – cet événement ne sera jamais envisagé, car tous les rattachements d'arbre sont supposés avoir été entièrement développés avant d'essayer de trouver une possibilité qui convienne;
- g) **REPEAT** – on ne cherchera jamais à identifier cet événement, car il aura été développé antérieurement;
- h) **GOTO** – cet événement est toujours évalué à VRAI et convient donc toujours;

- i) **EXPRESSIONS BOOLÉENNES** – une expression booléenne déclarée comme une possibilité convient si l'expression est évaluée à VRAI (voir le § D.6.7.2);
- j) **CLAUSES D'AFFECTION** – une clause d'affectation convient toujours;
- k) **ARBRES PARALLÈLES** – il ne sera jamais envisagé de convenance d'événements lors de l'exécution d'arbres parallèles, car ils devront avoir été entièrement développés avant d'essayer de trouver une possibilité qui convienne;
- l) **OPÉRATIONS DE TEMPORISATEUR** – **Start**, **Cancel**, **Suspend** et **Resume** - les opérations relatives aux temporisateurs conviennent toujours.

Remarque – Les événements spécifiés à la suite et au même niveau de retrait qu'un GO TO, qu'une clause d'affectation ou qu'une opération de temporisateur peuvent ne jamais être atteints.

D.7.4 *Algorithme complet*

Dans ce paragraphe:

- a) un verdict peut être l'un des suivants:
 - 1) succès, échec₁, échec₂, échec₃, non-concluant;
 - 2) aucun, qui est utilisé:
 - A) comme valeur spéciale retournée quand une feuille de sous-arbre est atteinte sans verdict;
 - B) comme valeur de verdict par défaut pour rattacher un sous-arbre qui ne comporte pas de verdict par défaut;
 - C) comme verdict rendu sur un événement, quand la colonne verdict est vide;
 - 3) une réunion ordonnée d'un ensemble ordonné . . . (ajoutée?);
- b) soit ÉVALUATION-DÉFAUT (R) une fonction:
 - 1) d'une référence de comportement (R);
 - 2) qui retourne un ensemble ordonné d'événements possibles;
 - 3) cet ensemble ordonné est défini comme suit:
 - A) soit d la référence du comportement par défaut associée à la description de comportement citée par R ;
 - B) s'il n'existe pas de d , retourner l'ensemble vide;
 - C) autrement, retourner la réunion ordonnée de:
 - i) tous les événements du premier niveau de retrait de la description de comportement auxquels fait référence d ;
 - ii) et évaluer ÉVALUATION-DÉFAUT (d);
- c) soit SÉLECTION (E) une fonction:
 - 1) d'un ensemble ordonné E d'événements possibles;
 - 2) qui retourne un événement de son ensemble d'entrée ou la valeur spéciale *erreur*;
 - 3) qui est défini comme suit:
 - A) soit E' un ensemble de doublets { événement, événement } initialisé comme:

$$\{ \{ e_1, e_1 \}, \{ e_2, e_2 \} \dots \{ e_n, e_n \} \}$$
 où $e_1 \dots e_n$ sont les éléments de E ;
 - B) remplacer tout événement de E' de la forme:

$$\{ +\text{arbre}, e \}, \dots \forall e \text{ par } \{ t_1, e \} \dots \{ t_n, e \}$$
 où $t_1, \dots t_n$ sont les événements du premier niveau de retrait de l'arbre;

Remarque 1 – Il faut recommencer (éventuellement de façon récursive) jusqu'à ce qu'il ne reste plus aucun élément de la forme { +arbre, e }, car un arbre peut rattacher un sous-arbre à son premier niveau de retrait.

Remarque 2 – Le second élément du doublet suit la trace du nom d'événement original dans le ??? de ???. La fonction SÉLECTION retournera cet élément.

Remarque 3 – E' est un ensemble ordonné.
 - C) si un **ELAPSE** se trouve parmi les possibilités, alors calculer T = unité-temps (valeur du temps écoulé), autrement faire T = infini.

- D) pour chaque élément de $\{ x, e \}$ de $\underline{E'}$, faire ce qui suit:
 SI la condition suivante est satisfaite:
 i) SI une expression booléenne figure et qu'elle est satisfaite;
 ET que L'UNE des conditions suivantes est satisfaite:
 – x est PCO? OTHERWISE et tout événement ASP est prêt à ce PCO;
 – x est PCO? ASP, l'ASP étant prête, toute expression booléenne utilisant des paramètres de cette ASP étant satisfaite et toutes les contraintes (le cas échéant) étant satisfaites;
 – x est PCO! ASP, l'ASP pouvant être émise à ce PCO (c'est-à-dire que le contrôle de flux autorise l'expédition) et toutes les contraintes (le cas échéant) étant satisfaites;
 – x n'est ni? ni!
 ALORS SÉLECTION se termine et retourne l'événement.
- E) si une ASP est prête à un PCO et que cette ASP ne peut pas être régie par le contrôle de flux, ou quand le délai d'un temporisateur est expiré, SÉLECTION se termine et retourne «erreur»;
- F) autrement:
 i) si un délai supérieur à T s'est écoulé depuis la première entrée dans c(3)D, choisir l'événement ELAPSE qui a été utilisé pour calculer T;
 ii) autrement repartir dans c(3)D;
 c'est-à-dire attendre l'événement suivant (boucle d'occupation).
- d) soit EXÉCUTION(x) une fonction;
 1) d'un événement x;
 2) qui se termine sans retourner de valeur;
 3) définie comme suit:
 A) si x est ?OTHERWISE, abandonner l'événement actuel (?ASP) qui a provoqué le retour de la fonction SÉLECTION et;
 B) si x est PCO? ASP, prendre l'ASP du PCO, lier tous paramètres ou variables libres associés aux valeurs reçues et;
 C) lier toutes variables globales non liées en sorte que l'expression booléenne soit satisfaite (le cas échéant) et;
 D) si x est PCO! ASP, lier les paramètres à des valeurs qui satisfassent l'expression booléenne et émettre l'ASP au PCO;
 E) effectuer toutes les affectations (y compris les opérations relatives aux temporisateurs);
- e) soit ARBRE-EVAL une fonction
 1) des éléments suivants:
 – du niveau courant de retrait (L) qui désigne un ensemble d'événements possibles d'une description de comportement;
 – du verdict par défaut courant (V) qui indique, dans le cas d'un rattachement d'arbre, si un verdict a été rendu au point d'attache de l'arbre;
Remarque – Aucun signifie aucun verdict.
 – de l'ensemble ordonné des événements par défaut (\underline{D}) à utiliser pour interpréter l'arbre;
- 2) ARBRE-EVAL retourne une valeur telle que définie dans a);
 3) ARBRE-EVAL est défini comme suit:
 A) évaluer \underline{K} = ensemble des événements du niveau L;
 B) lancer SÉLECTION ($\underline{K} \cup \underline{D}$);
 C) si e(3)B retourne *erreur*, alors ARBRE-EVAL se termine (erreur);
 D) si e(3)B retourne un événement de la forme + *arbre*, alors:
 i) calculer $V = \text{ARBRE-EVAL}(L', V', \underline{D}')$ où
 – L' désigne le premier niveau de retrait de l'arbre attaché;
 – V' est le résultat de l'application du tableau présenté à la figure D-29/X.290, partie 2 à V et le verdict attribué à l'arbre;

- \underline{D} est la réunion ordonnée d'ÉVALUATION-DÉFAUT (ref-arbre) et de D.
 - ii) si V est différent de aucun, alors retourner V;
 - E) autrement:
 - lancer EXÉCUTION(e) où e est l'événement retourné par e(3)B;
 - si e a un verdict, calculer V comme résultat de l'application du tableau présenté à la figure D-29/X.290, partie 2 à V et verdict affecté à l'événement, retourner V.
 - F) calculer L = niveau de retrait suivant, en tenant compte de tout **GOTO**;
 - G) si L désigne un ensemble vide, alors terminer et retourner V, autrement, continuer avec e(3)A.
- 4) l'interprétation d'un test élémentaire est donnée par ARBRE-ÉVALUATION (L, aucun, D) où:
- L = le premier niveau de retrait du premier arbre de la description des comportements du test élémentaire;
 - D = ÉVALUATION-DÉFAUT (référence-test-élémentaire);
 - A) si ARBRE-EVAL retourne «erreur», la spécification est incomplète;
 - B) si le résultat est «aucun», alors la spécification du test élémentaire est incomplète;
 - C) autrement, ARBRE-EVAL retourne le verdict.

D.8 *Partie déclaration des contraintes*

Il est nécessaire de décrire en détail le codage des paramètres dans les PDU et les ASP. La description du codage des paramètres se fait à l'aide soit d'une méthode tabulaire (§ D.8.1.1), soit d'une méthode qui tire parti de la notation ASN-1 (§ D.8.3). Des valeurs particulières sont mentionnées dans la colonne Référence des contraintes des tableaux utilisés dans la partie dynamique (§ D.6.13).

D.8.1 *Codage des paramètres*

Une ASP ou une PDU peut être considérée comme une liste de paramètres. Toutefois, du fait de l'importance du codage des PDU pour les tests, il peut être plus approprié de ne pas faire abstraction du codage des PDU et de considérer chacune d'elles comme une liste de champs, tels que «longueur», «type» et «valeur» du paramètre.

D.8.1.1 *Méthode tabulaire*

Si une valeur effective est affectée à chaque champ, la PDU ou l'ASP réelles peut être représentée par une liste de valeurs. Il est en outre possible de construire un ensemble de listes, dont chaque élément est une liste possible de valeurs prises parmi toutes les combinaisons. Cet ensemble peut alors, pour chaque PDU ou ASP, être représenté par un tableau. Le formulaire de tableau des déclarations de contraintes affectant une PDU est présenté à la figure D-34/X.290, partie 2.

Contraintes sur une PDU		
Nom de la PDU: <Nom de PDU>	Nom de la contrainte: <Nom de contrainte>	
Nom du champ	Valeur	Commentaire
.	.	.
.	.	.
.	.	.
<Nom du champ>	<Valeur>	<Commentaire>
.	.	.
.	.	.
.	.	.

FIGURE D-34/X.290, partie 2

Formulaire des contraintes sur une PDU

Remarque – Dans le texte ci-après, un tableau d'ASP ou d'ALP correspond à chaque tableau de PDU. Les formulaires de tableaux relatifs aux ASP ou ALP sont parfaitement similaires à ceux des PDU, sauf que l'abréviation PDU des champs du tableau doit être remplacée par ASP ou ALP.

Chaque entrée de la colonne «nom du champ» doit avoir été déclarée dans la déclaration de la PDU (ou de l'ASP ou ALP). Les valeurs affectées à chaque champ doivent être du type spécifié dans la déclaration de la PDU (ou de l'ASP ou ALP).

Quand une contrainte comporte un petit nombre de champs, ou quand le nombre de contraintes est réduit, la version compactée du formulaire de tableau des contraintes, présentée à la figure D-35/X.290, partie 2, peut être utilisée.

Contraintes sur une PDU				
Nom de la PDU: <Nom>				
Nom de la contrainte	Nom du champ			Commentaire
	<Nom de champ> ₁	...	<Nom de champ> _n	
<Nom de contrainte> ₁	<Valeur> _{1,1}	...	<Valeur> _{1,n}	<Commentaire> ₁
<Nom de contrainte> ₂	<Valeur> _{2,1}	...	<Valeur> _{2,n}	<Commentaire> ₂
		...		
<Nom de contrainte> _m	<Valeur> _{m,1}	...	<Valeur> _{m,n}	<Commentaire> _m

FIGURE D-35/X.290, partie 2
Formulaire de tableau compact des contraintes

Dans l'exemple D-39, les noms de champs sont en tête de colonne, et les différentes PDU sur des lignes successives du tableau. Si les champs sont trop nombreux pour tenir dans le sens de la largeur de la plage, il est possible de le transposer verticalement.

Exemple D-39 – Etant donné une PDU X, comportant deux champs P1 et P2, de valeurs possibles 0 et 1, le tableau est celui présenté à la figure D-36/X.290, partie 2.

Les colonnes Références des contraintes de la partie dynamique peuvent alors contenir des entrées telles que X(S1) et X(S4).

D.8.1.2 Valeurs génériques

Les contraintes peuvent être paramétrées à l'aide de valeurs génériques. Elles seront présentées dans le format montré à la figure D-37/X.290, partie 2. Là encore, une version compacte du format peut être utilisée. Elle est présentée à la figure D-38/X.290, partie 2.

Contraintes sur une PDU			
Nom de la PDU: X			
Nom de la liste	Nom du champ		Commentaire
	P1	P2	
S1	0	0	
S2	0	1	
S3	1	0	
S4	1	1	

FIGURE D-36/X.290, partie 2
Exemple de PDU comportant deux champs

Valeur générique		
Nom de la PDU: <Nom de PDU>	Nom générique: <Nom générique>	
Nom du champ	Valeur	Commentaire
.	.	.
.	.	.
.	.	.
<Nom du champ>	<Valeur>	<Commentaire>
.	.	.
.	.	.
.	.	.

FIGURE D-37/X.290, partie 2
Formulaire des contraintes sur une PDU

Valeur générique				
Nom de la PDU: <Nom de PDU>	Nom générique: <Nom générique>			
Nom de la liste	Nom du champ			Commentaire
	<Nom de champ> ₁	...	<Nom de champ> _n	
<Nom de contrainte> ₁	<Valeur> _{1,1}	...	<Valeur> _{1,n}	<Commentaire> ₁
<Nom de contrainte> ₂	<Valeur> _{2,1}	...	<Valeur> _{2,n}	<Commentaire> ₂
		...		
<Nom de contrainte> _m	<Valeur> _{m,1}	...	<Valeur> _{m,n}	<Commentaire> _m

FIGURE D-38/X.290, partie 2
Formulaire de tableau compact de contraintes génériques

Ces concepts sont illustrés dans les exemples D-39 et D-40.

Exemple D-40 – le déclenchement de la PDU X (figure D-39/X.290, partie 2) dans un module de test peut être effectué comme suit: (X(S1), X(S2), X(S3), X(S4), X(S5(0)), X(S5(1)) ou X(S5(Var)) où Var est une variable (voir le § D.5.6 et D.8.2) ou un nom de champ d'une PDU reçue (voir le § D.5.9) ou d'une expression.

Contraintes sur une PDU			
Nom de la PDU: X			
Nom de la liste	Nom du champ		Commentaire
	P1	P2	
S1	0	0	
S2	0	1	
S3	1	0	
S4	1	1	
S5(A)	1	A	

FIGURE D-39/X.290, partie 2
Exemple simple de contraintes paramétrées

Exemple D-41 – le mode de définition des valeurs génériques représentées dans l'exemple D-40 peut être étendu à un groupe de champs:

Etant donné la PDU Y, comportant trois champs, Q1, Q2 et Q3 de valeurs possibles 0 et 1, le tableau peut être présenté comme sur les figures D-40/X.290, D-41/X.290 et D-42/X.290, parties 2.

Contraintes sur une PDU				
Nom de la PDU: Y				
Nom de la liste	Nom du champ			Commentaire
	Q1	Q2	Q3	
T1	0	0	0	
T2(B1)	1	B1		
T3(B2)	B2		0	

FIGURE D-40/X.290, partie 2
Exemple de contraintes paramétrées (partie 1 de 3)

Valeur générique			
Nom de la PDU: Y	Nom générique: B1		
Nom de la contrainte	Nom du champ		Commentaire
	Q3	Q4	
C1	0	0	
C2	0	1	

FIGURE D-41/X.290, partie 2
Exemple de contraintes paramétrées (partie 2 de 3)

Valeur générique			
Nom de la PDU: Y	Nom générique: B2		
Nom de la contrainte	Nom du champ		Commentaire
	Q1	Q2	
D1	0	0	
D2	1	1	
D3	1	0	

FIGURE D-42/X.290, partie 2

Exemple de contraintes paramétrées (partie 3 de 3)

Le lancement de la PDU Y dans un module de test est alors effectué comme suit:

Y(T1), Y(T2(C1)), (T2(C2)), (Y(T2(D1))), etc.

Les tableaux définissant des valeurs génériques ne seront pas cités directement dans la partie relative au comportement principal, mais seulement par l'intermédiaire d'un nom de contrainte paramétrée (par exemple, B1 est utilisé pour Y(T2(B1)) dans l'exemple D-41).

D.8.2 Variables libres

D.8.2.1 Introduction

Dans de nombreux cas, il est nécessaire d'utiliser les valeurs reçues dans une PDU (par exemple, SRC-REF dans une CC) pour remplir le champ correspondant lors de l'expédition d'une autre PDU (par exemple, DST-REF dans DT). Bien que l'on puisse réaliser ceci en gardant trace de la valeur dans une variable de la partie dynamique, il peut être commode de la représenter de façon plus concise en utilisant une variable libre de la partie Contraintes. Une telle variable sera déclarée lors de la déclaration des contraintes, au début de la partie Contraintes. Les exemples D-40/X.290 et D-41/X.290, partie 2 ne comportent pas de variables libres. Dans ce cas, on peut représenter la valeur concernée par une variable libre qui est considérée comme globale à la partie Contraintes.

Ces variables libres doivent être déclarées en utilisant le formulaire de tableau présenté à la figure D-43/X.290, partie 2.

Une variable est liée à une valeur lorsque la PDU dans laquelle elle apparaît intervient comme un paramètre à une ASP reçue.

D.8.2.3 Variables dans des contraintes

Chaque fois qu'une contrainte s'applique, toute variable paraissant dans un champ de cette contrainte a ou prend la valeur réelle émise ou reçue.

- a) Si la variable est déjà liée à une valeur:
 - 1) cette valeur est envoyée; ou,
 - 2) cette valeur doit être la valeur reçue dans ce champ pour que la contrainte s'applique (autrement la contrainte ne s'applique pas).
- b) Si la variable est non liée:
 - 1) toute valeur du type approprié peut être envoyée; ou,
 - 2) la variable devient liée à la valeur reçue, du fait de l'application de la contrainte.

Une application typique de cette caractéristique peut être la conservation d'adresses ou de références de connexion d'une PDU à l'autre. Ces adresses ou références n'entrent pas dans l'objet de la plupart des tests, mais il est important que la même valeur figure dans des PDU différentes.

Variables libres		
Nom	Type	Commentaire
.	.	.
.	.	.
.	.	.
<Nom>	<Type>	<Commentaire>
.	.	.
.	.	.
.	.	.

FIGURE D-43/X.290, partie 2
Formulaire de tableau de variables libres

D.8.2.4 Conventions

Les conventions présentées dans le tableau D-1/X.290, partie 2, s'appliquent aux entrées des tableaux de contraintes.

TABLEAU D-1/X.290, partie 2
Conventions relatives aux entrées des tableaux de contraintes

Sens	Valeur du champ ou du paramètre	
	-	?
Lancé par le testeur (!)	Il n'y aura pas de codage explicite de ce champ ou paramètre.	La valeur de ce paramètre n'a aucune influence sur le résultat du test élémentaire et peut donc être quelconque à condition d'être autorisée par la norme* de service ou de protocole concernée.
Reçu par le testeur (?)	Il n'y aura pas de codage explicite de ce champ ou paramètre.	Le testeur n'a pas besoin d'opérer de vérification complémentaire de ce champ ou paramètre.

Remarque – Le concepteur d'un test élémentaire doit prendre garde au fait que si un paramètre ASN.1 a une valeur par défaut, une IUT est alors en droit de coder explicitement cette valeur par défaut, ou d'omettre le paramètre du codage (Recommandation X.209).

D.8.2.4.1 Comparaison de la configuration d'une chaîne de caractères

Dans une chaîne de caractères, un ? à la place d'un caractère signifie que tout caractère unique est accepté. Quand le symbole ? est lui-même nécessaire dans la chaîne, ceci doit être indiqué en le faisant précéder du caractère spécial \ . Le caractère \ sera lui-même écrit \ .

Dans une chaîne de caractères, un * signifie qu'aucun ou n'importe quel nombre de caractères est acceptable. Cet * pourra correspondre à la plus longue séquence de caractères possible, conforme à la configuration spécifiée dans les caractères qui encadrent le caractère *. (Quand le caractère * est lui-même nécessaire dans la chaîne, ceci sera indiqué en le faisant précéder du caractère spécial \).

D.8.3 Méthode modulaire ASN.1

D.8.3.1 Domaine d'application de la méthode

Cette méthode suppose que les descriptions ASN.1 des paramètres de PDU ou d'ASP sont disponibles et les utilisent comme base appropriée de description des contraintes.

D.8.3.2 Description de la méthode modulaire ASN.1

La méthode modulaire ASN.1 est applicable quand les paramètres d'une PDU ou d'une ASP sont définis dans la partie Déclarations en ASN.1. Cette méthode offre les principales possibilités suivantes:

- a) **Spécification:** spécification (partielle) de valeurs ASN.1. Normalement, les PDU seront correctes, mais n'importe quelles valeurs ASN.1 peuvent être construites et utilisées.
Remarque - ASN.1 nécessite des valeurs explicites, alors que la méthode modulaire autorise les «sans importance».
- b) **Dénomination:** ces valeurs peuvent être nommées, pour pouvoir s'y référer dans la partie dynamique.
- c) **Annotation:** les valeurs peuvent être annotées en utilisant une forme étendue des notations de type ASN.1. Les codages des valeurs peuvent être spécifiés, en sorte de pouvoir distinguer deux valeurs identiques codées différemment. Des contraintes peuvent être imposées aux composants de valeurs en utilisant des variables libres ou les conventions «-» et «?» exactement comme cela se fait pour les valeurs de champs dans la méthode tabulaire.
- d) **Remplacement:** de nouvelles valeurs peuvent être construites à partir de valeurs existantes en remplaçant des composants.

D.8.3.3 Spécification des valeurs

Pour spécifier une valeur ASN.1, les informations complémentaires suivantes seront fournies:

- a) Un nom.

Il est requis pour identifier la valeur.

Pour spécifier des valeurs ASN.1 longues et complexes, il est parfois pratique de n'imposer de contraintes qu'à certains composants, choisis, d'une valeur. Ces composants peuvent être définis en traçant un itinéraire dans la déclaration de type ASN.1 correspondant à la valeur, cet itinéraire identifiant les éléments de chaque niveau par un nom, puis en utilisant des points pour séparer les noms d'éléments.

- b) Le type ASN.1 de la valeur.

Ceci est requis à des fins documentaires, en sorte que la valeur ASN.1 puisse être associée à une définition de type ASN.1 appropriée se trouvant dans la norme de protocole concernée.

Un certain nombre d'abréviations et de restrictions sont imposées aux notations de type ASN.1 en sorte que les définitions de types ASN.1 soient utilisables sur le plan documentaire. Ces abréviations et restrictions sont énumérées ci-après. Certaines constituent les modifications par rapport à la grammaire donnée dans l'annexe F de la Recommandation X.208.

La notation suivante est utilisée:

```
<Non-Terminal> ::=  
    . . .  
    | <Nouvelles-Prods>
```

où <Non-Terminal> est défini dans l'annexe F de la Recommandation X.208 et . . . représente cette définition. <Nouvelles prods> représente les extensions TTCN.

- 1) Il est permis d'omettre le <type> d'une combinaison <identificateur> <Type>. Les productions suivantes sont ajoutées à l'annexe F de la Recommandation X.208:

<Type-Element> ::=
 ...
 | <Type-Elément-TTCN>
 <Type-Elément-TTCN> ::=
 <Type-Nommé-TTCN>
 <Type-Nommé-TTCN> ::=
 <identificateur>
 | <Type-nommé>

- 2) Dans un <Type-Choix>, seul le type du choix opéré doit être utilisé. Mais, le mot clé CHOICE peut être conservé pour rappeler au lecteur qu'un choix a été opéré. La production suivante remplace la production relative à <Type-Choix> de l'annexe F de la Recommandation X.208:

<Type-Choix> ::=
 <Type-Nommé-TTCN>
 | CHOICE <Type-Nommé-TTCN>

- 3) Le <Type-Sélection> ne doit pas être utilisé, le type réel sélectionné étant utilisé à la place. <Type-Sélection> est supprimé de la liste des <Types-Incluses> de l'annexe F de la Recommandation X.208.
- 4) Le <Type-Séquence-De> est modifié pour permettre la dénomination des valeurs réelles à envoyer. La production suivante remplace celle relative aux <Type-Séquence-De> de l'annexe F de la Recommandation X.208:

<Type-Séquence-De> ::= SÉQUENCE DE
 { <Liste-Types-Séquence-De> }
 <Liste-Types-Séquence-De> ::=
 ...
 | <Liste-Types-Séquence-De>
 <Liste-Type-Séquence-De>
 <Type-Types-Séquence-De> ::=
 <Type> . (numéro)

Tous les types apparaissant dans une <Liste-Types-Séquence-De> doivent être identiques. Les numéros d'une <Liste-Types-Séquence-De> doivent être distincts. Ils sont utilisés pour indexer les types.

- c) Une spécification des contraintes de codage de la valeur.

Si tout codage légal est acceptable, la spécification des contraintes de codage de la valeur doit être omise. Autrement, les spécifications de codage comprendront deux parties:

- 1) Une spécification de l'<Identificateur> apparaissant dans le codage de la valeur. L'<Identificateur> doit consister en un nombre entier d'octets.

<Identificateur> ::= [<Valeur-Identificateur>]
 <Valeur-Identificateur> ::=
 <chaîne-hexa>
 | <chaîne-binaire>
 | <variable>

Il n'est pas obligatoire que la <Valeur-Identificateur> soit correcte au regard de la spécification de type ASN.1. Autrement dit, des PDU de codage non correct peuvent être spécifiées.

- 2) Une spécification de la *<longueur>*, à utiliser dans le codage de la valeur. La longueur doit être un nombre entier d'octets.

<Identificateur> ::= [*<Valeur-Identificateur>*]

<Valeur-Identificateur> ::=

| *<chaîne-hexa>*
 | *<chaîne-binaire>*
 | *<variable>*
 | LI
 | SD
 | LD
 | LD *<numéro>*
 | IN

LI spécifie que tout codage légal de la longueur correcte peut être utilisé.

SD spécifie que le type de longueur *Définie courte* apparaîtra dans le codage.

LD spécifie que le type de longueur *Définie longue* apparaîtra dans le codage. La longueur sera complétée au *<nombre>* d'octets si (*nombre*) figure.

IN spécifie que le type de longueur *Indéfinie* peut apparaître dans le codage.

- 3) Une spécification de la valeur proprement dite.

<Valeur> ::= [*<Valeur-Nommée-TTCN>*]

<Valeur-Nommée-TTCN> ::=

| *<Valeur-Nommée>*
 | *<référence-valeur>*
 | ?
 | –
 | *<in>* *<identificateur>*₁
 | REPLACE *<identificateur>*₂
 | BY *<Valeur-Nommée-TTCN>*

<in> ::= IN | *<vide>*

<Valeur incluse> ::=

...
 | *<Valeur-Nommée-TTCN>*

Les valeurs sont spécifiées avec la notation de valeurs standard ASN.1, avec les extensions suivantes:

- A) Il peut être fait référence aux valeurs définies au § D.5.4 «Paramètres d'une suite de tests» ou ailleurs dans la partie Contraintes.
- B) Les valeurs *<Chaînes-Caractères>* peuvent être représentées en mettant la chaîne entre double guillemets. A l'intérieur de ces guillemets, un couple de doubles guillemets consécutifs sera utilisé pour représenter le caractère double guillemet.
- C) Un «?» est utilisé pour spécifier les valeurs «sans importance». Un ? apparaissant dans une valeur qui doit être émise par le testeur peut prendre toute valeur légale dans ce contexte (conformément à la norme de service ou de protocole concernée). Un testeur n'a pas besoin de vérifier les valeurs reçues spécifiées comme ?.
- D) Un – est utilisé pour spécifier que le codage de la valeur ne figurera pas.
- E) L'opération de remplacement donne la possibilité de remplacer des composantes de valeurs existantes par des valeurs ASN.1 (avec les contraintes correspondantes imposées aux types et aux codages), et de produire par là de nouvelles valeurs, avec la signification *remplacer le composant de nom <identificateur>*₁ de la valeur *<identificateur>*₁ par la *<Valeur-Nommée-TTCN>*.

Ces informations seront fournies dans le format présenté à la figure D-44/X.290, partie 2. Dans ce cas, le tracé des séparations des quatre dernières colonnes est facultatif.

Déclaration des contraintes sur une valeur ASN.1				
Type ASN.1	Identificateur	Valeur	Valeur	Commentaire
.
.
.
<Nom-Contrainte-Valeur-ASN.1> <Type ASN.1>	<Identificateur>	<Longueur>	<Valeur>	<Commentaire>
.
.
.

FIGURE D-44/X.290, partie 2

Déclaration des contraintes sur une valeur ASN.1

Exemple D-42 – cet exemple montre comment se référer à des valeurs de PDU de la partie dynamique. Définissons une PDU appelée APDU:

```
APDU ::= SÉQUENCE {
    INTÉGRER
    BOOLEAN
}
```

et la variable N. Voici un exemple d'occurrence d'APDU:

```
Attendue PDU ::= SÉQUENCE {
    INTÉGRER [10] [LI] N
    BOOLEAN [ID] [LI] TRUE
}
```

Nous pouvons alors écrire dans la description du comportement:

?DATAind [(Données-Utilisateur PDU-Attendue) AND (N = 27)]

APPENDICE I

(à la Recommandation X.290, partie 1)

Applicabilité des méthodes de test aux protocoles OSI *

I.1 Introduction

Les protocoles de la couche physique et de contrôle d'accès au support n'entrent pas dans le cadre de la présente Recommandation. Aucune des quatre méthodes définies dans cette Recommandation n'est destinée à être appliquée à ces protocoles.

I.2 Protocoles de liaison de données

Pour tester les protocoles de liaison de données, il faut prendre en considération les points suivants:

- a) La méthode de test monocouche locale est applicable uniquement si la frontière de la couche physique de l'IUT est accessible. En pratique, ceci a peu de chance d'être le cas, sauf lors du test d'un prototype ou de la conception d'un algorithme avant sa réalisation sous la forme d'un matériel ou d'un microprogramme.
- b) Comme le service physique n'est pas un service de bout en bout, les méthodes de test monocouches réparties, coordonnées et à distance sont applicables uniquement quand le testeur inférieur externe est connecté au SUT par une liaison unique et non par un réseau.
- c) Les méthodes de test monocouches réparties, coordonnées et à distance sont applicables uniquement si le testeur inférieur peut être réalisé en sorte qu'il ait un contrôle sur les primitives du service physique (ou, ce qui est peut-être plus réaliste, sur les PDU physiques et de liaison de données). Ceci peut être difficile pour certains types de sous-réseaux.
- d) La plupart des protocoles des sous-réseaux réels ont été définis avant la définition des services physique et de liaison de données. Les frontières des services situés au-dessous du service de réseau sont donc généralement non disponibles ou mal définies. On considère en général que le contrôle et l'observation pour des tests de la couche liaison de données doivent être interprétés en termes de PDU plutôt que d'ASP. Cela n'empêche pas certains systèmes de présenter une frontière de service de liaison de données ou une interface similaire, indépendante de la technologie.

Si le test monocouche de la mise en œuvre d'un protocole de liaison de données n'est pas possible, des méthodes multicouches ou monocouches encastrées devraient être envisagées.

I.3 *Protocoles de réseau*

Pour les protocoles de réseau, les méthodes de test à utiliser dépendent de la configuration: l'IUT est-elle un système d'extrémité ou un système-relais ouvert?

Comme les services assurés en dessous du service de réseau sont en général non disponibles ou mal définis, on considère que le contrôle et l'observation exercés pour des tests de mise en œuvre de la couche réseau doivent être interprétés en termes de PDU plutôt que d'ASP.

A noter qu'avec certaines technologies de sous-réseau, plus de trois protocoles sont nécessaires pour assurer le service de réseau. Chacun de ces protocoles peut être testé séparément ou en une combinaison quelconque avec des protocoles adjacents.

Si l'on considère la couche comme un tout, les primitives de service abstraites de réseau et de liaison de données sont contrôlables et observables. Pour les systèmes d'extrémité, les quatre méthodes de tests monocouches (non encastrées) sont donc applicables; mais comme le service de liaison de données n'est pas de bout en bout, le testeur inférieur doit être connecté au SUT pour pouvoir appliquer les méthodes de tests monocouches réparties, coordonnées et à distance.

Les méthodes de test en boucle et transversal sont applicables aux tests des systèmes-relais de la couche réseau.

I.4 *Protocole de transport*

Toutes les méthodes de test abstraites définies dans la présente Recommandation sont applicables aux tests de conformité au protocole de transport.

I.5 *Protocole de session*

Toutes les méthodes de test abstraites définies dans la présente Recommandation sont applicables aux tests de conformité au protocole de session.

Pour un grand groupe de systèmes, il sera approprié de tester le protocole de session en combinaison avec des protocoles de présentation et d'application. Les tests du protocole de session devraient donc être normalement effectués de l'une des deux façons suivantes:

- a) comme mise en œuvre monocouche, ou en combinaison avec des protocoles de niveaux inférieurs, pour tester la fourniture d'un service de session général capable de prendre en charge diverses applications différentes; les méthodes de test monocouches réparties ou coordonnées devraient être appropriées;
- b) en combinaison avec des protocoles de présentation et d'application, pour tester la mise en œuvre dans un contexte d'application spécifique; les méthodes de test monocouches encastrées, à distance ou réparties devraient être appropriées.

I.6 *Protocoles de présentation et d'application*

I.6.1 *Commentaires généraux*

Les tests de conformité peuvent être spécifiés de façon abstraite en termes de primitives de service, indépendamment de toute notion de point d'accès aux services associée à ces tests. Les tests peuvent ainsi être spécifiés en termes de primitives de service d'application, à condition que soit définie une certaine correspondance entre ces primitives et les effets réels qui peuvent être observés ou contrôlés. L'observation et le contrôle des primitives de service peuvent être indirects, du fait de la nature de leur correspondance avec des effets réels, mais des tests spécifiés en termes de telles primitives peuvent être exécutés, sous réserve que cette mise en correspondance soit possible. Il est difficile de trouver un exemple de primitives du service d'application qui ne pourraient jamais avoir de réalisations observables et contrôlables; si de telles primitives existaient, il semble probable qu'elles n'auraient jamais dû être définies comme des primitives.

Il est admis que, dans certaines circonstances, des normes ou des recommandations de protocole d'application peuvent spécifier des conditions portant sur des effets réels qui ont été obtenus du fait d'échange d'éléments de protocole (ceci est en particulier évident dans le transfert et la manipulation de travaux (JTM)). Toutefois, ces conditions portant sur des effets réels doivent être parfaitement séparées des conditions normales de conformité au protocole, et éventuellement formulées dans des normes ou recommandations (peut-être fonctionnelles) séparées. Certaines de ces conditions de conformité «hors protocole» peuvent être testables au moyen des méthodes de test abstraites d'usage général définies dans la présente Recommandation mais, en général, elles nécessiteront des méthodes de test spécifiques à une application, qui n'entrent pas dans le cadre de la présente Recommandation.

I.6.2 *Contrôle de l'association*

Le contrôle de l'association s'articule de façon inusuelle en ce sens qu'il comprend trois phases, dont la seconde est définie par un autre ASE. Dans certains systèmes, des primitives du service de contrôle de l'association peuvent être observées et contrôlées. Si c'est le cas, il devrait être possible de tester le protocole de contrôle d'association (éventuellement en combinaison avec le protocole de présentation) isolément de tout autre ASE. Pour les tests isolés du contrôle d'association, il faudra utiliser un ASE fictif. N'importe laquelle des méthodes de test pourra être utilisée, y compris la méthode coordonnée associée à un protocole de gestion de tests défini comme ASE fictif. Toutefois, de tels tests isolés sont de valeur limitée, car ils portent uniquement sur la machine protocole, sans tester la correspondance entre les syntaxes abstraites et de transfert. Cet aspect peut uniquement être testé pour un ASE particulier. L'utilisation d'un ASE fictif pour tester le contrôle d'association ne couvrira en général pas la mise en correspondance des syntaxes, comme ce serait le cas avec des ASE réels. Il faudrait donc donner la priorité au test du contrôle d'association en combinaison avec le protocole de présentation et un autre ASE réel, en appliquant les méthodes de test monocouches, encadrées, à distance ou réparties. Autrement dit, il convient d'insister sur le test des opérations du contexte d'application.

Le protocole de contrôle d'association ne fait pas l'objet de condition de conformité hors protocole.

I.6.3 *Terminal virtuel (VT)*

Le protocole de terminal virtuel peut être utilisé de trois façons: de terminal à hôte VT, de terminal à terminal et d'hôte VT à hôte VT. Dans les trois cas, les primitives du service VT doivent être observables et contrôlables sur une proportion de systèmes suffisante pour rendre réaliste leur utilisation dans une spécification de test. Les hôtes VT peuvent présenter des interfaces du service de VT, permettant la mise en œuvre de n'importe quels processus d'utilisateur: une telle interface peut être utilisée par une réalisation d'un testeur supérieur. Par ailleurs, les terminaux devraient très probablement exercer le contrôle et l'observation des primitives de service VT de façon seulement indirecte, à travers une interface utilisateur qui pourrait être très différente de celle du service VT. Toutefois, certains terminaux peuvent offrir un accès direct à une interface du service VT.

Les méthodes de test monocouches à distance et réparties, sont donc applicables aux tests des protocoles VT.

I.6.4 *Transfert, accès et gestion de fichiers (FTAM)*

En pratique, l'observation et le contrôle des primitives de service sont différentes pour les initiateurs et les répondeurs FTAM. En général, elles peuvent être observées et contrôlées lors des tests des initiateurs, mais pas lors de ceux des répondeurs. Le problème avec les répondeurs FTAM est qu'il n'existera pas d'effet associé à leurs primitives de service, et que l'observation et le contrôle de ces primitives seront extrêmement spécifiques au système. La raison en est que, bien que «l'utilisateur» du service FTAM soit, dans un répondeur, le système de fichiers virtuel, dans le système à tester, les effets du VFS ne peuvent être vus qu'au travers du système de fichiers réel. La méthode de test monocouche à distance est donc applicable aux initiateurs et aux répondeurs FTAM, mais la méthode de test monocouche répartie n'est applicable qu'aux initiateurs FTAM.

Les tests relatifs à la gestion des fichiers peuvent, si nécessaire, être effectués de façon autonome et ne dépendent pas des unités fonctionnelles de lecture et d'écriture qui sont réalisées. Les tests relatifs au transfert et à l'accès aux fichiers dépendent, par contre, de ces unités fonctionnelles qui sont réalisées.

Les tests relatifs au transfert et à l'accès aux fichiers pour des systèmes à lecture seule, ou à lecture/écriture sont relativement simples. Il est possible de tester les systèmes à lecture/écriture en leur transférant des fichiers qui sont ensuite relus. Bien qu'en général rien ne soit prévu pour empêcher des modifications locales de ces fichiers entre la lecture et l'écriture, on peut empêcher de telles modifications locales pendant la durée de l'exécution de la suite de tests en maintenant les autres utilisateurs à l'écart du système. Il est possible de tester les systèmes à lecture seule en préchargeant l'ensemble requis de fichiers par des moyens locaux et en exécutant ensuite des tests de lecture à l'aide de FTAM. Lors des tests de systèmes à écriture seule, les verdicts rendus sur les résultats doivent être fondés sur une interprétation hautement spécifique au système (éventuellement subjective) de ce que sont ces résultats (par exemple, enregistrement détaillé des événements de test observés).

Une autre façon d'envisager cette situation est d'admettre que FTAM ne comporte pas de conditions de conformité hors protocole concernant les effets réels qui peuvent se présenter comme résultats d'échanges de PDU. Ces conditions peuvent faire l'objet de tests conduits par des méthodes de test générales, à condition que ces méthodes comprennent l'observation des primitives du service FTAM (c'est-à-dire que l'exécution des tests comprenne l'observation des effets réels associée à ces primitives). Pour que les effets réels soient observés lors des tests, il faudra fournir au laboratoire de test une description détaillée de la correspondance entre les primitives de service et les effets réels. Cela équivaut à exiger la définition complète du système de fichiers réel. Il doit donc être fait référence à ces informations dans le PIXIT, car elles sont plus abondantes que celles que l'on souhaiterait inclure explicitement dans le PIXIT ou dans la PICS.

Une des conditions de conformité est qu'une réalisation FTAM soit, à des fins de tests, utilisable dans ses attributs Usage privé ou Qualification légale. En outre, d'autres attributs sont difficiles ou pratiquement impossibles à tester. Par exemple, les attributs relatifs à la protection des données et à la sécurité entrent dans cette catégorie, car les informations nécessaires à les tester ne seront probablement pas disponibles. En outre, la possibilité d'utiliser les attributs de fichier pour verrouiller un fichier contre un accès non OSI sera probablement impossible à vérifier, car le fait que le laboratoire de test ne peut pas accéder à un tel fichier n'indique rien en ce qui concerne les possibilités d'un utilisateur disposant d'autorisations d'accès plus étendues.

I.6.5 *Transfert et manipulation de travaux (JTM)*

Concernant l'observation et le contrôle des primitives de service, la situation est similaire à celle de FTAM. Pour les initiateurs JTM, les méthodes de test monocouches à distance et réparties seront donc applicables. Pour les répondeurs JTM, la méthode de test monocouche répartie peut être utilisée uniquement si les effets réels associés aux primitives de service peuvent être observés. Ceci impliquera la connaissance d'une correspondance qui peut être complexe. Toutefois, la norme de protocole JTM impose aux réalisateurs de déclarer quels sont ces effets et de rendre certaines informations lisibles par l'homme. Normalement, ces effets concernent divers documents qui sont créés et mis en circulation. Il peut ainsi être possible d'utiliser pour les initiateurs JTM les méthodes de test monocouches réparties et à distance.

Les primitives d'état ou de demande sont facilement testables. Les rapports d'avancement d'un travail peuvent être observés.

Une méthode de test multisystème est nécessaire pour tester les principaux aspects de JTM. Tous les systèmes sauf un peuvent être des testeurs (inférieurs) et plusieurs testeurs (inférieurs) logiquement distincts peuvent être réalisés dans le même système réel. D'un point de vue abstrait, il doit y avoir plus d'un testeur (inférieur) externe. Par exemple, les tests concernant les situations de dérangement doivent couvrir la situation dans laquelle un système est en panne et les autres systèmes censés revenir à la normale. Il peut également être désirable de tester un ensemble de systèmes plutôt qu'un système unique. Il doit donc y avoir au minimum deux nouveaux ensembles de méthodes de test, l'un concernant un SUT unique et deux testeurs (inférieurs) et l'autre concernant deux SUT et un testeur (inférieur) unique. De telles méthodes de test multisystèmes n'entrent pas dans le cadre de la présente Recommandation.

Remarque – Des conditions similaires concernant les méthodes de test multisystèmes existent pour MHS.

I.6.6 *Protocole de messagerie*

Il existe des suites de test CCITT pour trois protocoles: celui du service de messagerie de personne à personne (IPMS) (protocole P2), celui du service de transfert de messages (MTS) (protocole P1) et celui du service de transfert fiable (protocole RTS). Deux configurations de base des systèmes doivent en outre être prises en considération.

La première est la «configuration de test de systèmes d'extrémité» qui peut être utilisée pour tester P2, P1 et RTS. La seconde est la «configuration de test d'agent de transfert de messages relais» qui peut être utilisée pour tester les aspects relais du protocole P1.

La suite de tests du protocole P2 utilise un PCO externe à la frontière située entre la couche Agent d'utilisateur et le MTS et un PCO à la frontière supérieure de l'IUT. Mais comme le protocole P2 n'inclut pas de définitions d'ASP, il a été nécessaire de définir des ASP «fictives» pour décrire de façon formelle la suite de tests abstraite.

La suite de tests du protocole P1 pour les tests de systèmes d'extrémité utilise un PCO externe à la frontière située entre la couche Transfert de messages et le RTS et un PCO à la frontière supérieure de l'IUT. Mais le test de la remise à plusieurs destinataires nécessite plusieurs agents d'utilisateur, qui à leur tour nécessitent plusieurs PCO à la frontière supérieure de l'IUT. La suite de tests du protocole P1 pour le test de relais utilise deux PCO externes à la frontière située entre la couche Transfert de messages et le RTS.

La suite de tests du RTS utilise un PCO externe à la frontière située entre la couche de Transfert fiable et la couche Session et un PCO à la frontière supérieure de la couche Agent d'utilisateur, à l'intérieur du SUT. La suite de tests du RTS inclut également un troisième point d'accès aux services dans la description des tests élémentaires: il est situé à l'intérieur du SUT, entre la couche Transfert de messages et le RTS; il est utilisé pour rendre la description plus claire et il ne s'agit pas d'un PCO.

La méthode de test monocouche répartie est donc utilisée pour les tests de systèmes d'extrémité des protocoles P1 et P2; la méthode de test monocouche encastrée répartie est utilisée pour les tests du RTS; la méthode de test transversale est utilisée pour le test des aspects relais du protocole P1. Dans tous les cas de test de systèmes d'extrémité, la méthode de test monocouche à distance est également applicable.

Un dernier point est que les suites de tests abstraites des protocoles P1 et P2 comprennent des tests de codage et de décodage X.409.

Remarque – Les suites de tests auxquelles il est fait référence ici s'appliquent à des mises en œuvre de la version 1984 de la série X.400 de Recommandations du CCITT.

I.6.7 *Engagement, concurrence et reprise (CCR)*

Pour des raisons similaires à celles données au § I.6.2 pour le contrôle d'association, le CCR doit être testé dans un contexte d'application, comme ceci est requis pour JTM. Les méthodes de test monocouches, encastrées, à distance et réparties sont donc applicables.

La spécification des protocoles CCR comporte une condition visant à préserver la sécurité des données à la suite d'incidents, la réserve étant formulée que ce n'est pas réalisable en pratique, du fait que certains états d'erreur (par exemple, une bombe sur l'installation) sont trop catastrophiques. Pour pouvoir tester ladite condition visant à préserver la sécurité des données à la suite d'incidents, le réalisateur doit déclarer dans le PICS ou dans le PIXIT, quels sont les états d'erreur qui n'affectent pas la préservation de la sécurité des données.

I.6.8 *Présentation*

Les primitives de service sont potentiellement observables et contrôlables dans la même mesure que pour les couches des niveaux inférieurs. Les quatre méthodes de test monocouches (non encastrées) sont donc théoriquement applicables. Toutefois, comme expliqué plus haut pour le contrôle d'association, le test du protocole de présentation isolément d'un ASE est d'une valeur limitée, car il porterait uniquement sur la machine protocole, sans tester l'aspect le plus intéressant de la couche Présentation, à savoir la correspondance entre les syntaxes abstraite et de transfert. Le test d'un protocole de présentation encastré entre le contrôle d'association et un autre ASE est donc préférable. Sont donc applicables les méthodes de test monocouches encastrées, à distance et réparties.

I.6.9 *Syntaxes de transfert*

Les syntaxes de transfert (par exemple, ASN.1 ou X.409) sont, en ce qui concerne la conformité, assez différentes des Recommandations * de protocole OSI *. En général, il n'y aura pas de tests de conformité aux règles de codage d'une syntaxe de transfert, indépendants du protocole d'application utilisant ces règles.

A noter qu'il y a, par exemple, des conditions de conformité dans la Recommandation concernant les règles de codage de base ASN.1. La condition selon laquelle «lorsque plusieurs possibilités de codage sont fournies comme options proposées à l'expéditeur, les destinataires conformes doivent prendre en charge toutes ces possibilités», requièrent bien sûr des test de conformité spécifiques ASN.1. Néanmoins, les règles de codage de base ASN.1 seront toujours testées avec le protocole de présentation, et les méthodes de test choisies seront appropriées à ce protocole.

I.7 *Protocoles de gestion*

Comme la gestion du système et la gestion de l'application résident dans la couche Application, les commentaires généraux ci-dessus, qui s'appliquent aux protocoles d'application, s'appliquent également aux protocoles de gestion du système et de l'application. En particulier, les tests élémentaires de ces protocoles de gestion peuvent être spécifiés en termes de primitives de service de gestion, à condition qu'une certaine correspondance ait été définie entre les primitives du service de gestion et les effets réels qui peuvent être observés ou contrôlés. Si de telles primitives de service et une telle correspondance existent, alors les tests peuvent être exécutés par la méthode de test répartie. Autrement, les tests peuvent être effectués en utilisant la méthode de test à distance.

Comme avec les autres protocoles d'application, le transfert de PDU d'un protocole de gestion de système peut, en outre, être testé en utilisant les méthodes de test décrites dans la présente Recommandation. L'application des tests de conformité à d'autres aspects des services de gestion d'application et du système et les activités associées «hors protocole» des entités de gestion est, toutefois, hors du cadre de la présente Recommandation.

Du fait des conditions de conformité «hors protocole» associées aux protocoles de gestion de systèmes, les tests de conformité à ces protocoles ne peuvent donc pas être entièrement réalisés au moyen des méthodes décrites dans la présente Recommandation. Par exemple, le fonctionnement correct de la fonction de gestion du système de modification des informations d'un annuaire ne peut pas être testé simplement par l'échange de PDU.

En outre, les fonctions de gestion de système ne sont pas simplement en relation avec la couche Application, mais également avec les activités des protocoles de niveaux inférieurs. Le test du protocole de gestion de système doit donc être exécuté à l'aide d'une mise en œuvre des protocoles de niveaux inférieurs qui a déjà été testée.

Les tests des protocoles de gestion de couche dépendent d'abord de l'existence desdits protocoles; supposant qu'ils existent, il serait possible de les tester au moyen des méthodes de test décrites dans la présente Recommandation.

I.8 *Protocoles en mode sans connexion*

Comme chaque méthode de test décrite dans la présente Recommandation est définie en termes d'observation et de contrôle d'ASP et de PDU et non en termes de connexions, toutes ces méthodes sont applicables aux tests des protocoles en mode sans connexion, compte tenu des restrictions s'appliquant à chaque couche.

APPENDICE II
(à la Recommandation X.290, partie 1)

Index des définitions de termes

<i>Terme</i>	<i>Paragraphe</i>	<i>Terme</i>	<i>Paragraphe</i>
ALP	3.8.5	Rapport de test de conformité au protocole	3.7.8
ASP	3.8.4	Rapport de test de conformité du système	3.7.7
ASP (N)	3.8.4	Réalisateur de l'équipement de test	3.8.14
Capacités d'une IUT	3.4.5	Réalisation à tester	3.4.1
Client	3.4.12	Réalisation conforme	3.4.10
Comparabilité (des résultats)	3.7.2	Reproductibilité (des résultats)	3.7.1
Conditions de conformité dynamique	3.4.3	Résultat prévu	3.7.4
Conditions de conformité statique	3.4.4	Résultats	3.7.3
Corps du test	3.6.8	Revue de conformité statique	3.5.7
Déclaration de conformité d'une mise en œuvre de protocole	3.4.6	SCTR	3.7.7
Déclaration de conformité du système	3.4.11	Suite de tests	3.6.12
Épilogue	3.6.9	Suite de tests abstraite	3.6.16
Évaluation de conformité	3.5.10	Suite de tests abstraite paramétrée	3.6.24
Événement de test	3.6.11	Suite de tests abstraite sélectionnée	3.6.20
Événement de test inopportun	3.7.11	Suite de tests d'interconnexion de base	3.6.19
Événement de test syntaxiquement non valide	3.7.10	Suite de tests de conformité	3.6.18
Événement de test valide	3.7.9	Suite de tests exécutable	3.6.17
Événement imprévu	3.7.5	Suite de tests exécutable sélectionnée	3.6.21
Formulaire PICS	3.4.7	Suite de tests exécutable paramétrée	3.6.25
Formulaire PIXIT	3.4.9	Suite de tests générique	3.6.15
Groupe de tests	3.6.14	SUT	3.4.2
IUT	3.4.1	Système à tester	3.4.2
Journal de conformité	3.7.15	Test actif	3.5.1
Laboratoire de test	3.4.13	Test d'interconnexion de base	3.5.5
Méthode de test à distance	3.8.12	Test de capacités	3.5.6
Méthode de test abstraite	3.6.1	Test de comportement	3.5.8
Méthode de test coordonnée	3.8.11	Test de conformité	3.5.9
Méthode de test répartie	3.8.10	Test élémentaire	3.6.13
Méthodes de test externes	3.8.9	Test élémentaire abstrait	3.6.3
Méthodes de test locales	3.8.8	Test élémentaire abstrait paramétré	3.6.22
Méthodologie de test abstraite	3.6.2	Test élémentaire exécutable	3.6.4
Module de test	3.6.10	Test élémentaire exécutable paramétré	3.6.23
Objet du test	3.6.5	Test élémentaire générique	3.6.6
PCO	3.8.1	Test encastré	3.5.4
PCTR	3.7.8	Test multicouche	3.5.3
PICS	3.4.6	Test passif	3.5.2
PIXIT	3.4.8	Testeur inférieur	3.8.2
Point de contrôle et d'observation	3.8.1	Testeur réel	3.8.13
Préambule	3.6.7	Testeur supérieur	3.8.3
Primitive du service (N) abstraite	3.8.4	Verdict	3.7.6
Primitive locale abstraite	3.8.5	Verdict «échec»	3.7.13
Procédures de coordination de tests	3.8.6	Verdict «non concluant»	3.7.14
Protocole de gestion de tests	3.8.7	Verdict «succès»	3.7.12

APPENDICE III

(à la Recommandation X.290, partie 2)

Exemples de directives à l'intention des concepteurs de formulaires

III.1 Abréviations

Les exemples suivants utilisent les abréviations: «m» pour obligatoire, «c» pour conditionnel, «o» pour optionnel, «n» pour négociable et «-» pour non applicable, tel que défini au § 7.1.7.

III.2 Exemple de PDU

Nom de la PDU	Référence du paragraphe	Prise en charge	
		Emission	Réception
PDU-1	a.b.c	m	m
PDU-2	a.b.d	o	m
PDU-3	a.b.e	c	-

III.3 Exemple de paramètres

Paramètre	Référence du paragraphe	Plage de valeurs autorisées	Valeurs prises en charge	Type de prise en charge
PARM-1	x.y et p.q	illimité		n
PARM-2	f.g.h	1 → 10		o
PARM-3	f.i	4		c
PARM-4	w.x.y.z	type A		m

Une variante de cet exemple serait celui d'un temporisateur, avec éventuellement une colonne supplémentaire pour les unités.

III.4 Exemple de service pris en charge

Option	Paragraphe	Pris en charge par			
		Deux boîtes de conserve et une corde humide	Pigeon voyageur	Perception ultra-sensorielle	Ultrasons
Chuchotement	4	m	—	o	m
Echo	5	n	o	c	n
Sifflement	6	o	—	c	m

APPENDICE IV

(à la Recommandation X.290, partie 2)

Exemple de choix de méthodes de test abstraites

IV.1 *Raisonnement pour la couche Transport*

Objet: tester une entité de transport (IUT monocouche) d'un système ouvert réel.

Hypothèse 1: le système ouvert réel est utilisé pour plusieurs applications.

Hypothèse 1.1: toutes les applications utilisent le service de transport OSI, via une interface définie localement et accessible.

Hypothèse 1.2: toutes les applications utilisent le service de transport OSI via une interface du service de session, définie localement et accessible, la frontière du service de transport étant accessible.

Hypothèse 1.3: toutes les applications utilisent le service de transport OSI via le service de session OSI, mais ni la frontière du service de transport, ni la frontière du service de session ne sont accessibles.

Hypothèse 2: le système ouvert réel est utilisé pour une seule application.

Hypothèse 2.1: toutes les frontières de couches sont accessibles via des interfaces définies localement.

Hypothèse 2.2: aucune frontière de couche n'est accessible - le système ne fournit aucune interface, sauf à l'utilisateur final (c'est le cas des produits monolithiques télétext et MHS).

Hypothèse 2.3: aucune frontière de couches n'est accessible - le système ne fournit aucune interface - l'utilisateur final ne peut même pas accéder à la frontière entre les parties OSI et non OSI du processus d'application.

Les méthodes de test externe s'appliquent comme suit à ces hypothèses:

1.1	directement	CS	ou	DS		
1.2	encadrée via la couche Session	CSE	ou	DSE		
1.3		RS				
2.1	directement	CS	ou	DS	ou	LS
2.2	encadrée via toutes les couches supérieures	CSE	ou	DSE		
2.3		RS				

Conclusions: les méthodes de test CS, CSE, DS, DSE, DS et LS s'appliquent toutes à la couche Transport. Les méthodes de test encadrées peuvent être utilisées via seulement la couche Session ou via toutes les couches supérieures.

Priorité: les hypothèses 1.1, 1.2 et 2.2 paraissent être les cas les plus courants (respectivement méthode d'accès direct, la méthode d'accès par session et les produits monolithiques télétext et MHS).

Il faudrait donc donner la priorité aux méthodes de test CS, CSE et DSE.

IV.2 Principales méthodes de test à utiliser pour la couche Transport

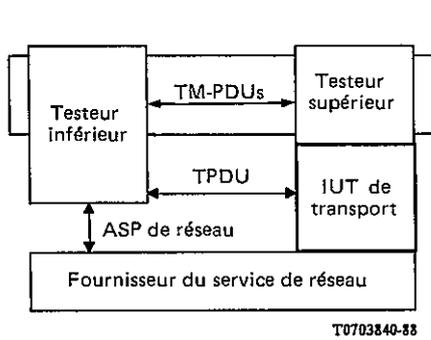


FIGURE IV-1/X.290, Partie 2

La méthode de test CS

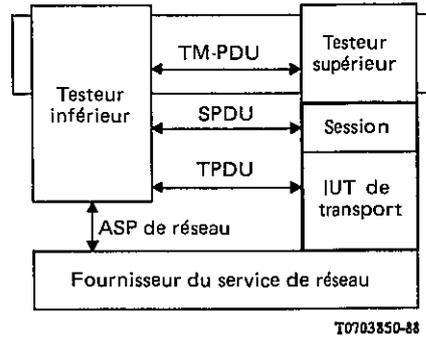


FIGURE IV-2/X.290, Partie 2

La méthode de test CSE

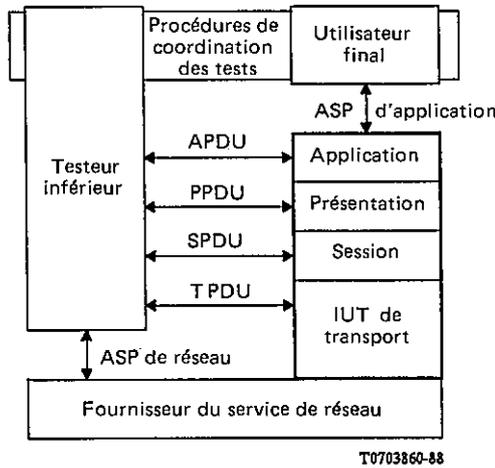


FIGURE IV-3/X.290, Partie 2

La méthode de test DSE

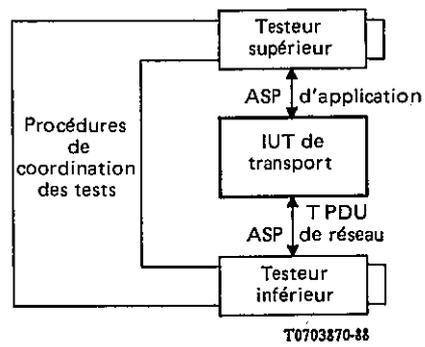


FIGURE IV-4/X.290, Partie 2

La méthode de test LS

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects informatiques généraux des systèmes de télécommunication