



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

X.745

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(11/93)

**RÉSEAUX DE COMMUNICATION DE DONNÉES ET
COMMUNICATIONS ENTRE SYSTÈMES OUVERTS
GESTION OSI**

**TECHNOLOGIE DE L'INFORMATION –
INTERCONNEXION DE SYSTÈMES
OUVERTS – GESTION DES SYSTÈMES:
FONCTION DE GESTION DES TESTS**

Recommandation UIT-T X.745

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT), (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI. Le texte de la Recommandation X.745 de l'UIT-T a été approuvé le 16 novembre 1993. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10164-12.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

© UIT 1995

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives 2
3	Définitions..... 3
4	Abréviations 5
5	Conventions..... 6
6	Spécifications 6
7	Modèle pour la fonction de gestion des tests 7
8	Définitions génériques 15
9	Définitions des services..... 22
10	Unités fonctionnelles de gestion de systèmes 28
11	Protocole 29
12	Relations avec d'autres fonctions 34
13	Conformité 34
	Annexe A – Définition d'informations de gestion 37
	Annexe B – Exemples 49

Résumé

La présente Recommandation | Norme internationale spécifie un modèle et des objets gérés permettant de lancer des tests sur des ressources distantes. Ces tests peuvent être soit pilotés avec la possibilité de les suivre, de les suspendre et de les reprendre au cours de leur exécution, soit non pilotés, leurs résultats n'étant alors retournés qu'une fois le test achevé.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIE DE L'INFORMATION – INTERCONNEXION DE
SYSTÈMES OUVERTS –
GESTION DES SYSTÈMES: FONCTION DE GESTION DES TESTS**

1 Domaine d'application

La présente Recommandation | Norme internationale définit, dans le cadre de la Rec. X.700 du CCITT | ISO/CEI 7498-4, une fonction de gestion des systèmes interactive qui peut être utilisée par un processus d'application dans un environnement de gestion centralisé ou décentralisé. Elle définit une fonction qui se compose de définitions génériques, de services et d'unités fonctionnelles. Cette fonction se situe dans la couche application OSI de la Rec. X.200 du CCITT | ISO/CEI 7498 et répond au modèle ISO 9545. La Rec. X.701 du CCITT | ISO/CEI 10040 décrit le rôle des fonctions de gestion de systèmes.

La présente Recommandation | Norme internationale

- établit les besoins des utilisateurs pour la présente Recommandation | Norme internationale;
- établit un modèle de test qui relie les définitions de services et les définitions génériques fournies par cette fonction aux besoins des utilisateurs;
- définit des classes d'objets génériques, des types d'attribut, d'action et de notification, ainsi que des blocs et des paramètres présentés conformément à la Rec. X.722 du CCITT | ISO/CEI 10165-4;
- spécifie les conditions de compatibilité imposées aux autres normes qui utilisent ces définitions génériques;
- définit le service fourni par la fonction;
- spécifie le protocole nécessaire pour fournir le service;
- définit les relations entre ce service et les opérations et notifications des informations de gestion des systèmes;
- spécifie la syntaxe abstraite permettant d'identifier et de négocier l'unité fonctionnelle dans le protocole;
- définit les relations avec d'autres fonctions de gestion des systèmes;
- spécifie les conditions de conformité que vise à remplir l'application de la présente Recommandation | Norme internationale.

En revanche, la présente Recommandation | Norme internationale

- ne définit pas la nature d'une éventuelle mise en œuvre destinée à assurer la fonction de gestion des tests;
- ne spécifie pas comment l'utilisateur de la fonction de gestion des tests en assure la gestion;
- ne définit pas la nature des interactions pouvant entraîner l'utilisation de la fonction de gestion des tests;
- ne spécifie pas les services nécessaires à l'établissement et à la libération, normale ou anormale, d'une association de gestion;
- ne précise pas le détail de tests ou de catégories de tests spécifiques;
- ne définit pas de cadre ou de méthodologie pour les tests de conformité.

2 Références normatives

Les Recommandations | Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations | Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations | Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le TSB de l'UIT-T tient à jour une liste des Recommandations de l'UIT-T actuellement en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.210 (1993) | ISO/CEI 10731:1993, *Technologie de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Conventions pour la définition des services OSI.*
- Recommandation X.701 du CCITT (1992) | ISO/CEI 10040:1992, *Technologie de l'information – Interconnexion de systèmes ouverts – Aperçu de la gestion de systèmes.*
- Recommandation X.731 du CCITT (1992) | ISO/CEI 10164-2:1992, *Technologie de l'information – Interconnexion de systèmes ouverts – Gestion des systèmes – Fonction de gestion d'états.*
- Recommandation X.732 du CCITT (1992) | ISO/CEI 10164-3:1992, *Technologie de l'information – Interconnexion de systèmes ouverts – Gestion des systèmes – Attributs pour représenter les relations.*
- Recommandation X.734 du CCITT | ISO/CEI 10164-5:1992, *Technologie de l'information – Interconnexion de systèmes ouverts – Gestion des systèmes – Fonction de gestion des rapports d'événements.*
- Recommandation X.720 du CCITT (1992) | ISO/CEI 10165-1:1992, *Technologie de l'information – Interconnexion de systèmes ouverts – Structure des informations de gestion: Modèle d'information de gestion.*
- Recommandation X.721 du CCITT (1992) | ISO/CEI 10165-2:1992, *Technologie de l'information – Interconnexion de systèmes ouverts – Structure des informations de gestion: Définition des informations de gestion.*
- Recommandation X.722 du CCITT (1992) | ISO/CEI 10165-4:1992, *Technologie de l'information – Interconnexion de systèmes ouverts – Structure des informations de gestion: Directives pour la définition des objets gérés.*

2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique

- Recommandation X.200 du CCITT (1988), *Modèle de référence pour l'interconnexion de systèmes ouverts pour les applications du CCITT.*
ISO 7498:1984, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base.*
- Recommandation X.208 du CCITT (1988), *Spécification de la syntaxe abstraite numéro un (ASN.1).*
ISO/CEI 8824:1990, *Technologie de l'information – Interconnexion de systèmes ouverts – Spécification de la notation de syntaxe abstraite numéro un (ASN.1).*
- Recommandation X.209 du CCITT (1988), *Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1).*
ISO 8825:1990, *Technologie de l'information – Interconnexion de systèmes ouverts – Spécification de règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1).*
- Recommandation X.700 du CCITT, *Définition du cadre général de gestion pour l'interconnexion de systèmes ouverts (OSI) dans les applications du CCITT.*
ISO/CEI 7498-4:1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 4: Cadre général de gestion.*
- Recommandation X.710 du CCITT (1991), *Définition du service commun de transfert d'informations de gestion pour les applications du CCITT.*
ISO/CEI 9595:1991, *Technologie de l'information – Interconnexion de systèmes ouverts – Définition du service commun d'informations de gestion.*

- Recommandation X.290 du CCITT (1991), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications du CCITT – Concepts généraux*.
ISO/CEI 9646-1:1991, *Technologie de l'information – Interconnexion de systèmes ouverts – Cadre général et méthodologie des tests de conformité – Partie 1: Concepts généraux*.

2.3 Références additionnelles

- ISO/CEI 9545:1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Structure de la couche application*.

3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

3.1 Définitions du modèle de référence de base

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.200 du CCITT | ISO 7498:

- a) système ouvert;
- b) gestion des systèmes.

3.2 Définition du cadre de gestion

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.700 du CCITT | ISO/CEI 7498-4:

- a) objet géré;
- b) entité d'application de gestion de systèmes.

3.3 Définitions du service commun de transfert d'informations de gestion (CMIS)

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.710 du CCITT | ISO/CEI 9595:

- a) attribut;
- b) élément de service commun de transfert d'informations de gestion;
- c) service commun de transfert d'informations de gestion.

3.4 Définitions relatives aux opérations distantes

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.219 du CCITT | ISO/CEI 9072-1:

- a) demandeur;
- b) exécutant.

3.5 Définitions de la vue d'ensemble de la gestion des systèmes

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.701 du CCITT | ISO/CEI 10040:

- a) agent;
- b) rôle d'agent;
- c) conformité dépendante;
- d) conformité générale;
- e) définitions génériques;
- f) classe d'objet géré;
- g) système (ouvert) géré;

- h) gestionnaire;
- i) rôle de gestionnaire;
- j) utilisateur-MIS;
- k) notification;
- l) type de notification;
- m) protocole d'application de gestion des systèmes;
- n) unité fonctionnelle de gestion des systèmes.

3.6 Définitions relatives au modèle d'information de gestion

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.720 du CCITT | ISO/CEI 10165-1:

- a) type d'attribut;
- b) confinement.

3.7 Directives pour la définition des objets gérés

La présente Recommandation | Norme internationale utilise le terme suivant défini dans la Rec. X.722 du CCITT | ISO/CEI 10165-4:

modèle.

3.8 Définitions de la fonction de gestion des rapports d'événements

La présente Recommandation | Norme internationale utilise le terme suivant, défini dans la Rec. X.734 du CCITT | ISO/CEI 10164-5:

discriminateur de retransmission d'événement.

3.9 Définitions des tests de conformité OSI

La présente Recommandation | Norme internationale utilise le terme suivant, défini dans la Rec. X.290 du CCITT | ISO/CEI 9646-1:

déclaration de conformité du système.

3.10 Définitions supplémentaires

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes sont applicables:

3.10.1 objets associés: Objets gérés autres que des objets gérés renvoyant à un test, des objets de test et objets gérés ayant des fonctionnalités de récepteur de demande d'action de test, qui sont mis à contribution dans un test.

3.10.2 test piloté: Test pour lequel sont créés, aux fins de contrôle et de pilotage, un ou plusieurs objets de test dont l'identificateur d'invocation de test a la même valeur. Les résultats du test sont fournis par des notifications émises par les objets de test ou sont disponibles sous forme de valeurs d'attributs de ces objets.

3.10.3 test gênant: Indication donnée concernant une invocation de test si le service fourni aux utilisateurs est interrompu au risque de l'être par suite du test.

3.10.4 objet géré renvoyant à un test: Objet géré utilisé pour faire référence aux fonctionnalités à tester.

3.10.5 test non gênant: Indication donnée concernant une invocation de test si le service fourni aux utilisateurs n'est pas interrompu par suite du test.

3.10.6 compte rendu demandé: Compte rendu des résultats de test obtenu en utilisant des attributs d'objets de test.

3.10.7 récepteur de demande d'action de test: Terme désignant la capacité d'un objet géré à intervenir sur une demande de test. L'objet géré peut avoir d'autres attributs, procédures et notifications que ceux qui sont exigés de lui en tant que récepteur de demande d'action de test.

Un objet géré disposant de cette fonctionnalité peut créer (puis supprimer) une instance d'une classe d'objet de test donnée (à la suite de la réception d'une demande de test piloté), représentant une invocation de test spécifique.

3.10.8 catégorie de test: Un ou plusieurs tests ayant les mêmes buts et des caractéristiques semblables. Une catégorie de test définit le comportement supplémentaire et les modèles nécessaires pour réaliser un test.

3.10.9 responsable de test: Gestionnaire qui établit la procédure de test.

3.10.10 exécution de test: Phase d'un test durant laquelle est déterminé le résultat.

3.10.11 invocation de test: Instance spécifique d'un test, à partir de son lancement jusqu'à sa fin.

3.10.12 objet de test: Objet géré existant uniquement le temps d'une invocation de test piloté et ayant des attributs, procédures et notifications relevant de cette instance de test.

3.10.13 exécution d'objet de test: Phase d'un test qui débute avec un passage à l'état test en cours et se termine par une transition de cet état à un autre état. Un objet de test peut se prêter à une ou plusieurs exécutions à la suite d'une seule demande de test.

3.10.14 procédures de test: Procédures de gestion concernant les tests.

3.10.15 exécuteur de test: Agent qui reçoit les demandes de test à exécuter.

3.10.16 demande de test: Demande adressée par le responsable de test à l'exécuteur, pour lancer une ou plusieurs invocations de test.

3.10.17 résultats de test: Informations générées au cours de l'exécution d'un test et mises à la disposition d'un système ouvert.

3.10.18 session de tests: Ensemble d'invocations de tests.

3.10.19 test non piloté: Test ne faisant l'objet d'aucune opération de surveillance ou de pilotage. Les résultats du test sont fournis dans une ou plusieurs réponses à la demande de test.

3.10.20 compte rendu non sollicité: Compte rendu des résultats de test obtenu à l'aide de notifications appropriées émises par un objet de test.

4 Abréviations

AO	Objet associé (<i>associated object</i>)
ASE	Élément de service d'application (<i>application service element</i>)
CMIS	Service commun de transfert d'informations de gestion (<i>common management information service</i>)
conf.	Confirmation
DN	Nom distinctif (<i>distinguished name</i>)
Id	Identificateur
ind.	Indication
MAPDU	Unité de données du protocole d'application de gestion (<i>management application protocol data unit</i>)
MORT	Objet géré renvoyant à un test (<i>managed object referring to test</i>)
PDU	Unité de données de protocole (<i>protocol data unit</i>)
dem.	Demande
RDN	Nom distinctif relatif (<i>relative distinguished name</i>)
rép.	Réponse
SMAE	Entité d'application de gestion de systèmes (<i>systems management application entity</i>)
SMAPM	Machine protocole d'application de gestion de systèmes (<i>systems management application protocol machine</i>)
TARR	Récepteur de demande d'action de test (<i>test action request receiver</i>)
TO	Objet de test (<i>test object</i>)

5 Conventions

La présente Recommandation | Norme internationale définit des services selon les conventions descriptives définies dans la Rec. UIT-T X.210 | ISO/CEI 10731. Dans son article 9, la définition de chaque service comporte un tableau qui donne la liste des paramètres des primitives du service. Pour une primitive de service donnée, la présence de chaque paramètre est décrite par l'une des valeurs suivantes:

- M Le paramètre est obligatoire (*mandatory*).
- (=) La valeur de ce paramètre est égale à celle du paramètre de la colonne de gauche.
- U L'utilisation de ce paramètre est une option de l'utilisateur du service.
 - Le paramètre est absent dans l'interaction décrite par la primitive en question.
- C Le paramètre est conditionnel. La ou les conditions sont définies par le texte décrivant le paramètre.
- P Paramètre soumis aux contraintes imposées par la Rec. X.710 du CCITT | ISO/CEI 9595.

NOTE – Les paramètres marqués «P» dans les tableaux de service de la présente Recommandation | Norme internationale sont mis en correspondance directe avec les paramètres correspondants de la primitive du service CMIS, sans modifier la sémantique ou la syntaxe des paramètres. Les autres paramètres sont utilisés pour construire une unité MAPDU.

6 Spécifications

La présente Recommandation | Norme internationale a pour objet de satisfaire les besoins de pilotage à distance des tests et fournit un cadre pour la spécification de tests sur les ressources d'un système ouvert. Les besoins pour la même fonctionnalité de test peuvent provenir de divers domaines évolués, tels que la gestion des anomalies ou des performances. Par exemple un test donné peut être utilisé pour générer des informations qui peuvent être ensuite exploitées pour vérifier une fonctionnalité, déterminer la cause d'une anomalie ou pour élaborer des statistiques de performances.

Un test consiste à faire fonctionner et à contrôler des systèmes ouverts, ou certains de leurs éléments, dans un environnement permettant d'obtenir des informations sur les fonctionnalités et/ou les performances du ou des systèmes testés.

Chaque test peut exiger la création d'un environnement nécessaire au test, au pilotage et au contrôle de l'exécution du test, ainsi qu'au rétablissement de l'environnement normal. Le pilotage d'un test doit permettre de l'interrompre, de le reprendre et de l'arrêter. Chaque test doit être identifié de manière unique afin, par exemple, de conserver la trace des données générées par le test.

Dans certains cas, il est nécessaire de spécifier des tests qui puissent être suspendus, repris et arrêtés dans certaines conditions prédéfinies.

Les parties de l'environnement système devant parfois être modifiées à des fins de test sont les suivantes:

- la connexion avec d'autres systèmes ouverts;
- la configuration des systèmes testés;
- la charge de travail requise des systèmes testés.

Il est nécessaire, dans certains cas, de planifier les tests. Cette planification doit être envisagée de façon périodique et aperiodique et doit être modifiable. Le test doit parfois être exécuté au moment le plus opportun pour le système utilisé.

Il peut être nécessaire de spécifier un test de sorte qu'il déclenche automatiquement quand apparaît un ensemble de conditions prédéfini (par exemple, un seuil est franchi) ou quand un événement précis est détecté.

Il est aussi nécessaire de créer des tests complexes à partir de tests élémentaires: par exemple, regrouper les résultats de plusieurs tests secondaires en un seul résultat ou mettre des tests en séquence pour détecter une anomalie dans une entité ayant un grand nombre de composants.

Il peut être nécessaire d'exécuter plusieurs tests pour répondre à un besoin spécifique. Il faut alors regrouper dans un seul résultat les résultats obtenus par chaque test. Il faudra parfois prévoir l'identification unique de cette série de tests.

Cette fonction est applicable à différentes méthodologies, par exemple: la méthode de test en boucle qui configure une ressource de telle sorte que les données envoyées soient ensuite reçues; les tests par introduction d'anomalies dans

lesquels des erreurs sont délibérément introduites afin de vérifier qu'elles sont résolues correctement; ou les tests automatiques émettant simplement une indication de réussite ou d'échec.

NOTE – Il est nécessaire pour le responsable des tests d'établir le rang de priorité d'un test donné. La présente Recommandation | Norme internationale ne répond pas à ce besoin.

7 Modèle pour la fonction de gestion des tests

Le présent article fournit la terminologie relative aux tests; il définit les aspects qui sont indépendants des catégories des tests ainsi que ceux qui sont propres à chaque catégorie de tests.

7.1 Environnement

L'exécution d'un test implique deux ou plus de deux processus d'application (utilisateurs MIS). Le test le plus simple n'exige que deux processus, un processus gestionnaire, le *responsable de test* qui lance le test, et un processus agent, l'*exécuteur de test* qui exécute le test à la demande du responsable du test. (Des exemples de tests impliquant plus de deux utilisateurs MIS sont donnés dans l'Annexe B.)

La Figure 1 illustre une instance d'invocation de test unique avec un seul sous-ensemble d'échanges possibles de protocole. Les objets gérés n'apparaissent pas. Par souci de simplicité, cette figure et les suivantes ne représentent pas toutes les réponses de protocoles, par exemple en cas d'utilisation d'un service confirmé de compte rendu d'événements.

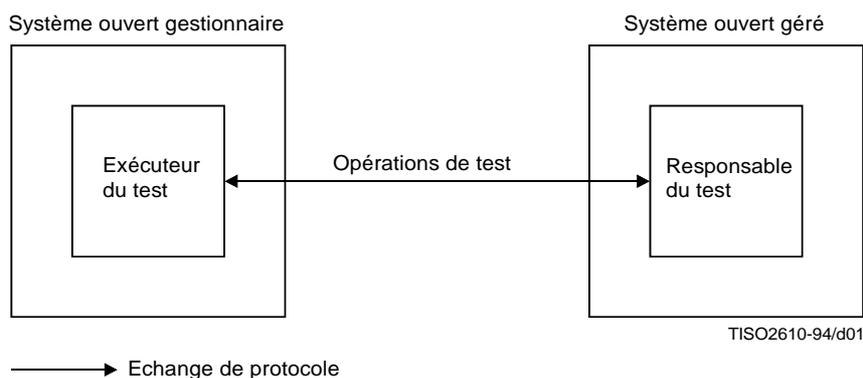


Figure 1 – Modèle de test

Une demande de test est adressée à un objet géré, par l'exécuteur de test, et disposant de la fonctionnalité appropriée pour recevoir une telle demande et pour y répondre. Cette fonctionnalité s'appelle fonctionnalité de récepteur de demande d'action de test (TARR). Les objets gérés qui font référence aux fonctionnalités à tester, les MORT, sont identifiés dans les demandes de test. Chaque test comprend un ou plusieurs MORT. Pour un test, la TARR peut faire partie de la fonctionnalité soit d'un MORT, soit d'un autre objet géré. Par exemple, la TARR peut faire partie d'un objet géré dont le rôle est expressément de recevoir des demandes de tests.

L'exécution d'un test peut inclure l'utilisation de protocoles de couche (N) ou de gestion de couche (N), ou d'autres mécanismes de gestion de systèmes spécifiques, ainsi que l'utilisation de protocoles de gestion de systèmes. Les résultats des tests peuvent permettre de déduire des informations sur le fonctionnement des ressources représentées par le ou les MORT.

Un test est *non piloté* ou *piloté*. Le comportement général d'un test est dicté par sa catégorie.

7.1.1 Tests non pilotés

Un test non piloté est un test qui n'est ni contrôlé ni piloté. Les résultats de test sont fournis en une ou plusieurs réponses à la demande de test. Pour les tests non pilotés, une invocation de test démarre au lancement de la demande et s'achève au retour de la réponse finale. Un test non piloté est modélisé à l'aide d'un ou de MORT et d'un objet géré doté de la fonctionnalité TARR. La Figure 2 illustre un exemple de test non piloté.

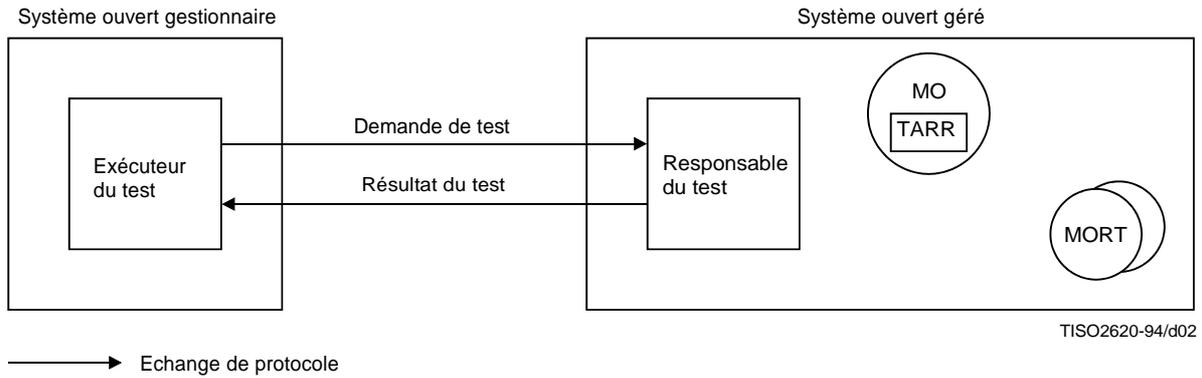


Figure 2 – Exemple de test non piloté

7.1.2 Tests pilotés

Un test piloté est un test pour lequel sont créés, aux fins de contrôle et de pilotage, un ou plusieurs objets de test (TO) dont l'attribut identificateur d'invocation de test a la même valeur. Les TO d'un test piloté sont instanciés à la suite d'une seule demande de test. Les résultats du test apparaissent sous forme de valeurs d'attributs des TO et/ou sont fournis dans des notifications émises par les TO. Ces notifications peuvent être envoyées à n'importe quel système ouvert sous forme de comptes rendus d'événements en utilisant, dans l'exécuteur de test, un discriminateur de retransmission d'événement approprié, tel que défini dans la Rec. X.734 du CCITT | ISO/CEI 10164-5. Les résultats des tests peuvent en outre être stockés en instanciant la procédure appropriée de l'exécuteur de test décrite dans la Rec. X.735 du CCITT | ISO/CEI 10164-6.

Un test piloté est modélisé à l'aide de TO, de MORT et d'un objet géré doté de la fonctionnalité TARR. La Figure 3 illustre un exemple de test piloté.

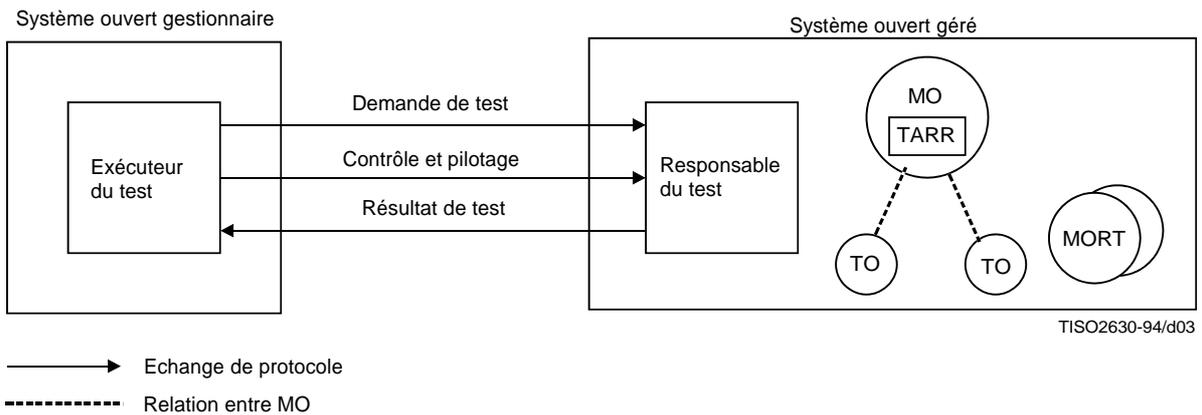


Figure 3 – Exemple de test piloté

Les TO détiennent des informations sur le test. Une seule demande de test peut créer un nombre quelconque de TO. Des relations peuvent exister entre les TO et le comportement d'un TO peut dépendre du comportement d'autres TO. Le nom du TO peut être attribué par le responsable de test ou par l'exécuteur de test. Ce dernier attribue la valeur de l'attribut identificateur d'invocation de test qui identifie l'invocation de test.

Les demandes de suspension, de reprise ou d'arrêt d'un test piloté sont adressées à l'objet disposant de la fonctionnalité TARR. Les TO concernés sont identifiés par un identificateur d'invocation de test ou un identificateur de session de test. Des demandes d'interruption prématurée de test, si elles sont autorisées, peuvent être adressées aux TO. Les spécifications d'un test peuvent décrire les conditions sous lesquelles il sera exécuté, suspendu, repris et arrêté. Ces conditions peuvent être régies soit par un mécanisme de planification, l'état d'un MORT, soit par la détection d'un événement spécifique.

7.2 Modèle fonctionnel

7.2.1 Lancement d'un test

La demande de lancement du ou des tests émise par le responsable de test à l'intention de l'exécuteur de test inclut toutes les informations nécessaires et peut comporter les éléments suivants:

- l'identification du ou des MORT;
- les informations propres à la catégorie du test;
- un délai d'exécution;
- l'identification d'un ou de plusieurs objets associés;
- un identificateur de session de test.

Pour les tests pilotés, le responsable de test peut également

- indiquer si un ou plusieurs tests sont demandés;
- indiquer les classes et, en option, les noms du ou des TO requis;
- inclure des informations concernant les valeurs initiales d'attributs pour le ou les TO.

Pour les tests non pilotés, la réponse à une demande de test acceptée contient les résultats des tests. Dans le cas des tests pilotés, cette réponse comporte les noms distinctifs du ou des TO, leur identificateur d'invocation de test et, en option, les valeurs initiales d'attributs. La réponse à une demande rejetée comporte une indication d'échec accompagnée de toutes les informations s'y rapportant.

Les TO créés suite à une demande de test piloté peuvent être des tests indépendants ou faire partie du même test composé. Le paramètre type de demande de test piloté est utilisé pour indiquer si les TO sont indépendants ou associés. Une demande de test indépendant lance plusieurs (un ou plus d'un) tests, chacun d'eux comprenant un seul TO. Une demande de test composé lance un seul test comprenant plus d'un TO associé. Une demande unique de test piloté peut lancer ou un seul test composé, ou un ou plusieurs tests indépendants. Un test piloté avec un seul TO est toujours un test indépendant.

7.2.2 Planification des tests

Le responsable de test peut ménager une fenêtre temporelle dans laquelle il voudrait que s'exécute un TO. S'il est possible à l'exécuteur de test de planifier le test dans le délai prévu, il renvoie une confirmation positive et peut préciser l'heure de lancement réelle (ou prévue) et l'heure de fin du test telles qu'il les a calculées ou prévues. Il peut également signaler toute modification de ces horaires. Si l'heure de fin effective dépasse l'heure de fin fixée ou si le lancement du test a lieu avant l'heure demandée, l'objet de test doit cesser s'il y a lieu d'exécuter le test et émettre une notification de non-respect de planification.

Le responsable de test peut ménager, à la demande, une fenêtre temporelle dans laquelle le TO doit être dans l'état en cours de test. L'heure de lancement est la première heure à laquelle le responsable de test souhaite que le TO entre dans l'état en cours de test et l'heure de fin est la dernière heure à laquelle le responsable de test souhaite qu'un TO sorte de cet état. Ces informations sont fournies dans le bloc fenêtre demandée.

Le responsable de test peut fixer l'instant auquel le TO entre dans l'état en cours d'initialisation. Ce peut être une heure absolue ou il peut dépendre de l'heure à laquelle le TO entrera dans l'état en cours de test. Ces informations sont fournies dans le bloc instant d'initialisation.

L'exécuteur de test peut fournir l'heure réelle de lancement et de fin pour les TO. Ces informations ne peuvent pas être directement modifiées par le responsable de test. L'heure réelle de lancement est l'heure à laquelle l'exécuteur de test s'attend à ce qu'un objet de test commence ou a commencé à s'exécuter. L'heure réelle de fin est l'heure à laquelle l'exécuteur de test s'attend à ce qu'un objet de test finit ou a fini de s'exécuter. Ces informations sont définies dans le bloc heure réelle de test.

La Figure 4 montre des exemples de planification des tests. Le premier exemple illustre la notion de fenêtre à la demande, le deuxième l'utilisation de la période de temporisation pour un test.

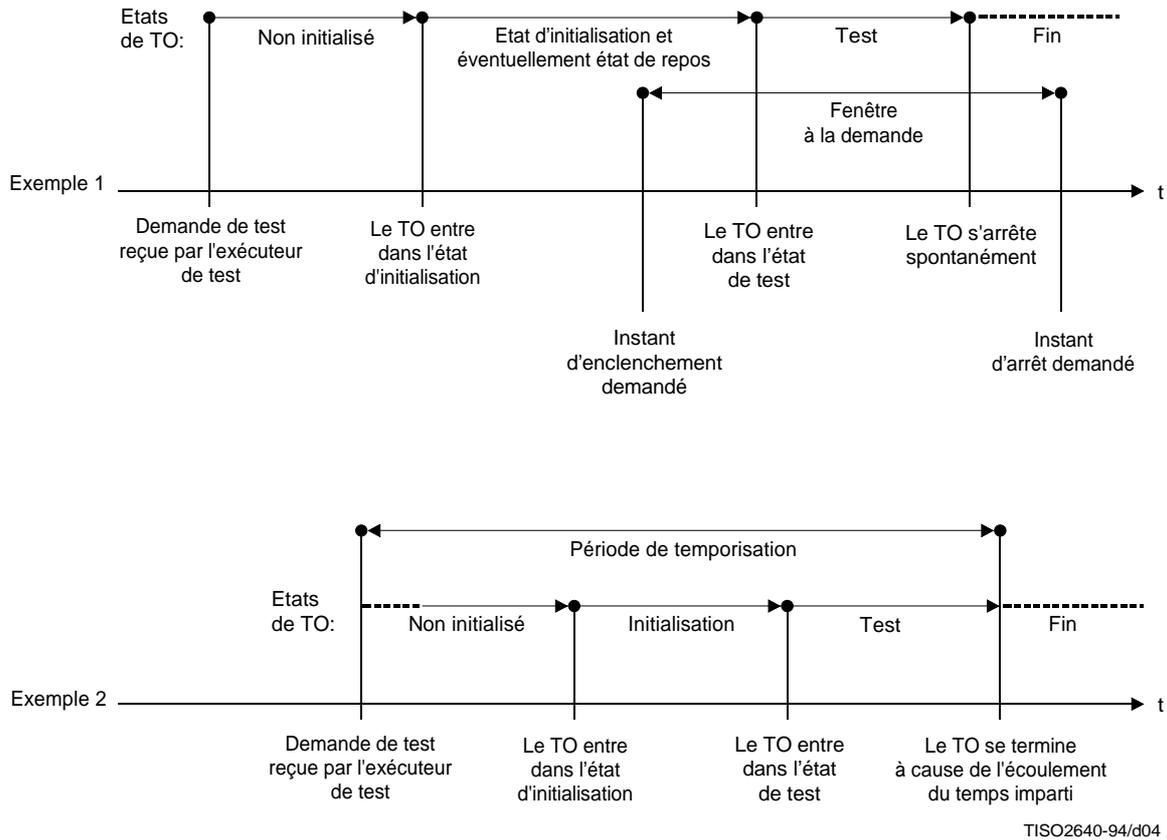


Figure 4 – Exemples de planification de test

7.2.3 Compte rendu des résultats de test

Dans le cas de tests non pilotés, les résultats de test sont rapportés dans une ou plusieurs confirmations données à la demande de test. La confirmation finale indique la fin du test et doit contenir le paramètre résultat de test qui ne doit être présent que dans la confirmation finale. Ce paramètre peut prendre une des valeurs suivantes: succès, échec, non concluant, hors des délais ou terminaison prématurée. L'interprétation de ce paramètre dépend du type de test requis. En outre, les confirmations peuvent comporter des informations relatives au type de test. Si le paramètre indique échec, la primitive de confirmation peut contenir des paramètres indiquant la nature du problème et les mesures correctives proposées.

Pour les tests pilotés, les résultats peuvent être obtenus auprès du TO par le responsable de test (réponse sollicitée), ou être fournis sous forme de notifications émises par le TO (réponse non sollicitée).

Dans le cas de compte rendu non sollicité, les résultats sont fournis par une ou plusieurs notifications émises par un TO. Une notification de résultat doit contenir l'identificateur d'invocation de test du test. L'identificateur de session de test doit également y figurer s'il apparaît dans la demande de test. Le TO doit indiquer la fin de l'envoi des comptes rendus d'exécution d'un test en incluant le paramètre résultat de test dans un compte rendu. Les comptes rendus des résultats peuvent contenir toute autre information conformément aux spécifications des notifications correspondant aux TO.

L'identificateur de session de test peut être utilisé pour la corrélation des résultats. L'algorithme permettant cette corrélation est en dehors du domaine d'application de la présente Recommandation | Norme internationale.

Un TO peut être défini de telle sorte qu'il conserve les résultats de tests sous forme de valeurs d'attributs, en remplacement ou en complément de l'émission de notifications. Le responsable de test peut alors extraire directement ces informations en lisant les attributs correspondants.

Que les résultats des tests soient sollicités ou non, les informations données par le paramètre résultat de test ne peuvent être fiables qu'en fin d'exécution des tests.

7.2.4 Suspension et reprise de test

Seul un test piloté peut être suspendu ou repris par une demande de suspension/reprise, adressée à l'objet qui a reçu la demande de test. L'aboutissement de la suspension ou de la reprise signifie que tous les TO du test sont suspendus/repris. Les tests sont identifiés dans les demandes de suspension/reprise à l'aide soit d'un identificateur d'invocation de test, soit d'un identificateur de session de test.

Si un identificateur de session de test est fourni dans une demande de suspension/reprise, les invocations de tests identifiés par la session doivent être suspendues/reprises au mieux.

Si un identificateur d'invocation de test est fourni dans une demande de suspension/reprise, tous les TO identifiés par le test doivent être suspendus/repris. Dans le cas contraire, un message d'erreur doit être renvoyé.

En réponse à une demande de suspension/reprise, la liste de toutes les invocations de test qui ont été suspendues/reprises doit être renvoyée.

Lorsqu'un test est suspendu, tous les TO passent dans l'état suspendu. Les définitions de TO indiquent si l'état de fonctionnement normal du ou des MORT est restauré au cours de la suspension. La définition de TO indique à quel point de son déroulement le test sera repris. Les états des TO sont indiqués dans la confirmation de la demande de reprise de test.

7.2.5 Terminaison de test

Un test peut se terminer spontanément ou sur demande. Les tests non pilotés ou pilotés peuvent se terminer spontanément, à la fin du test ou par suite de conditions anormales (par exemple, conflits de planification pour des tests pilotés). Seuls les tests pilotés peuvent être terminés par une demande de terminaison de test ou abandonnés par suppression de tous les TO ayant le même identificateur d'invocation de test.

La terminaison spontanée se produit lorsque des critères ou une anomalie prédéfinis sont vérifiés. Ces critères et certaines anomalies précises sont indiqués pour la catégorie de test ou la classe des TO. Pour les tests non pilotés, les résultats finaux des tests ou une indication d'échec doivent être renvoyés.

Une demande de terminaison adressée à l'objet qui a reçu la demande de test peut mettre fin à un test piloté. La terminaison réussie d'un test implique que tous les TO sont terminés. Les tests sont identifiés dans les demandes de terminaison soit par un identificateur d'invocation de test, soit par un identificateur de session de test.

Si un identificateur de session de test est fourni dans la demande de terminaison, les invocations de tests identifiés par la session doivent être terminées au mieux.

Si un identificateur d'invocation de test est fourni dans la demande de terminaison, tous les TO identifiés doivent être terminés. Dans le cas contraire, un message d'erreur doit être renvoyé.

En réponse à une demande de terminaison, la liste de toutes les invocations terminées suite à cette demande doit être renvoyée.

Lorsqu'un test est terminé, les TO du test exécutent une séquence de terminaison pouvant inclure l'émission d'un compte rendu des résultats du test, l'élimination des données superflues et l'arrêt des MORT et des objets associés. Si un compte rendu de résultat est émis et que le résultat de test n'est pas concluant, ce résultat doit indiquer terminaison prématurée. L'ordre de la ou des séquences de terminaison est spécifique au système et n'est pas défini par le test.

Un test piloté peut être abandonné en supprimant tous les TO ayant le même identificateur d'invocation de test. Les objets de test acceptant les demandes de suppression n'émettent aucun compte rendu de résultats de test.

7.3 Informations

7.3.1 Identificateur d'invocation de test

Tous les tests pilotés, indépendants ou composés, doivent avoir un identificateur unique appelé identificateur d'invocation de test. Les identificateurs d'invocation de test sont renvoyés dans la réponse à une demande de test. Tous les TO appartenant au même test doivent avoir le même identificateur d'invocation de test. Pour une demande de test

composé, le même identificateur d'invocation de test est attribué à tous les TO du test en question. Pour une demande de test indépendant, chaque TO représente un seul test. Dans ce cas l'identificateur d'invocation de test peut être le nom distinctif (DN) du TO.

La réponse à une demande de test piloté renvoie l'identificateur d'invocation de test de chaque test appelé par la demande. Pour une demande de test composé, un seul identificateur d'invocation de test est renvoyé. Les identificateurs des TO instanciés, suivis, éventuellement, de leur liste d'attributs, sont eux aussi fournis. Pour une demande de test indépendant, un identificateur d'invocation de test est renvoyé pour chaque TO instancié, suivi, éventuellement, du nom du TO et/ou de sa liste d'attributs.

L'identificateur d'invocation de test est obligatoire dans la classe d'objets TO.

7.3.2 Identificateur de session de test

Une session de test est un ensemble d'invocations de test. Une session de test est identifiée par un identificateur de session de test. S'il est utilisé, l'identificateur de session de test doit être attribué par le responsable de test et indiqué dans la demande de test. Le responsable de test peut utiliser à sa convenance les identificateurs de session de test. S'il figure dans la demande de test, le même identificateur de session de test doit également figurer dans tout TO instancié et dans les éventuelles notifications de résultat émises par les TO.

7.3.3 Etats de test

Des TO doivent prendre en charge un sous-ensemble des attributs d'état définis dans la Rec. X.722 du CCITT | ISO/CEI 10164-2 pour permettre de consulter l'état en cours d'un test.

Un TO peut présenter sept états:

- a) **Non initialisé** – Le TO est disponible, en attente de passer dans l'état en cours d'initialisation. Cette transition peut être commandée par l'attribut instant d'initialisation du bloc instant d'initialisation.
- b) **Inactif** – Le TO n'est pas encore passé dans l'état en cours de test, les conditions de lancement de test n'ayant pas encore été satisfaites. La transition vers l'état en cours de test peut être commandée par l'attribut heure réelle de début du bloc heure réelle de test.
- c) **En cours d'initialisation** – C'est la phase de préparation de l'environnement de test. Quand cet environnement est prêt, le TO passe soit dans l'état inactif, soit dans l'état en cours de test suivant que les critères de test sont satisfaits. Un test peut rester dans cet état pendant un certain temps, si, par exemple, il faut attendre que des objets gérés soient dans un état administratif ou opérationnel spécifique.
- d) **Suspendu** – Un TO peut être interrompu pour une demande de suspension. Dans cet état, il est inactif mais ses attributs sont visibles; les attributs accessibles en lecture peuvent être lus et les attributs accessibles en écriture peuvent être modifiés. Par exemple, pendant la suspension d'un test, les attributs de résultat peuvent être lus et les paramètres de planification peuvent être modifiés. Le test peut être repris par une demande de reprise, tel que spécifié par le comportement du TO.
- e) **En cours de test** – L'état en cours de test reflète la phase du TO pendant laquelle se déroulent les algorithmes et les mesures de test.
- f) **En cours d'achèvement** – L'environnement de test est démonté. Cette opération peut couvrir les activités nécessaires pour restaurer le ou les MORT, ou les ressources de test, dans leur condition précédant le test. Le démontage de l'environnement du test peut exiger un certain temps.
- g) **Désactivé** – Le TO est indisponible lorsqu'il devient totalement inutilisable à la suite d'une panne. Cet état peut être temporaire ou provoquer un arrêt anormal du test.

Certaines classes de TO ne présentent que certains de ces états. Toutefois, tous les TO doivent au moins avoir l'état en cours de test. Par exemple, un test n'exigeant pas d'environnement de test particulier peut se passer de temps d'installation et peut ne pas présenter l'état en cours d'initialisation ou l'état en cours d'achèvement. De la même façon, un test sans planification ne présente pas l'état inactif. Le Tableau 1 donne la correspondance entre les états définis ci-dessus et les attributs définis dans la Rec. X.731 du CCITT | ISO/CEI 10164-2. L'attribut d'état de commande n'est nécessaire que si le TO prend en charge l'état suspendu. L'attribut d'état de disponibilité n'est nécessaire que pour la planification.

Tableau 1 – Mise en correspondance des états de test

Etat de test	Etat opérationnel	ID d'état de procédure	ID d'état de commande	ID d'état de disponibilité
Non initialisé	Activé	Non initialisé	Non interrompu	Hors service
Inactif	Activé	(Vide) ou faisant rapport	Non interrompu	Hors service
En cours d'initialisation	Activé	En cours d'initialisation	Non interrompu	(Vide) ou hors service
En cours de test	Activé	(Vide) ou faisant rapport	Non interrompu	(Vide)
En cours d'achèvement	Activé	En cours d'achèvement (peut aussi faire rapport)	Non interrompu	(Vide)
Suspendu	Activé	(N'importe quelle valeur)	Suspendu	(N'importe quelle valeur)
Désactivé	Désactivé	(N'importe quelle valeur)	(N'importe quelle valeur)	(N'importe quelle valeur)

La Figure 5 montre un exemple de diagramme de transition d'états pour un TO prenant en charge la planification. La figure ne montre qu'une seule exécution de test.

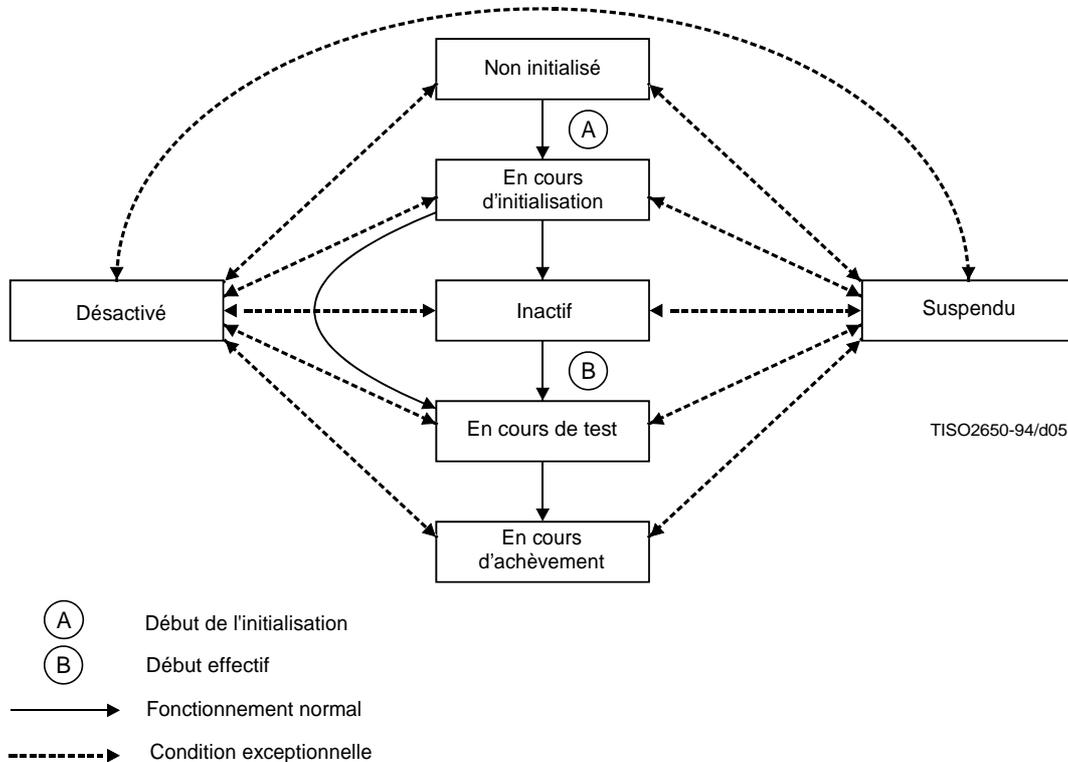


Figure 5 – Diagramme de transition d'états de test

7.3.4 Etape de test

Un TO peut avoir un bloc d'étape de test composé d'un attribut d'étape de test et d'un possible qualificateur d'étape de test. L'attribut d'étape de test peut refléter les étapes dans chaque état de test de TO. Cet attribut n'est accessible qu'en lecture; lorsqu'il est utilisé, il est défini par la classe de TO. Le qualificateur de test fournit des informations (chaîne graphique) sur les étapes du test telles que des conditions de disponibilité ('en attente d'événement externe' ou 'en attente de données de gestion'), des déclarations de procédures ('arrêt sur erreur', 'boucle sur erreur' ou 'boucle') ou des informations relatives au statut ('anomalie détectée'). Ces informations permettent au responsable de test de surveiller l'évolution du test et toute condition spéciale.

7.4 Spécification d'informations de test

Ce formulaire traite des informations de gestion qui, prises ensemble, spécifient un test. Les tests pilotés et les tests non pilotés peuvent faire partie de la même catégorie de test mais peuvent être présentés séparément.

Les éléments d'informations qui ne relèvent pas de la catégorie de test à présenter peuvent être omis.

7.4.1 Nom de la catégorie de test

Le nom de cette catégorie de test.

7.4.2 But de la catégorie de test

Description du test et des raisons possibles d'appeler le test.

7.4.3 Conditions relatives aux MORT

Description des MORT auxquels s'applique le test. (Peut inclure une liste d'objets gérés.)

7.4.4 Conditions relatives aux objets associés

Description des MORT auxquels s'applique le test. (Peut inclure une liste d'objets gérés.)

7.4.5 Conditions relatives à la fonctionnalité TARR

Conditions relatives à l'objet géré ayant la fonctionnalité TARR.

Les paragraphes 7.4.5.1 à 7.4.5.5 peuvent être répétés pour chaque service de demande de test applicable à cette catégorie de test.

7.4.5.1 Type de service de demande de test

Le nom du modèle de demande de test. (Indique le bloc de test piloté/non piloté qui doit être utilisé.)

7.4.5.2 Erreurs spécifiques

Indique chaque modèle de paramètre d'erreur spécifique qui spécifie une réponse à une erreur pour ce test et ce service.

7.4.5.3 Paramètre d'informations sur la catégorie de test

Les noms des modèles de paramètre spécifiant les valeurs possibles de ce paramètre.

7.4.5.4 Paramètre d'informations supplémentaires

Les noms des modèles de paramètre spécifiant les valeurs possibles de ce paramètre (tests non pilotés uniquement).

7.4.5.5 Pilotage de test

Indique chacun des services de pilotage possibles, axé sur l'objet géré pourvu de la fonctionnalité TARR, applicable à la catégorie (tests pilotés uniquement).

7.4.5.5.1 Erreurs spécifiques au pilotage de test

Indique chaque modèle de paramètre d'erreur spécifique qui spécifie une réponse à une erreur pour ce service.

7.4.6 Conditions relatives au TO

Indique chaque classe de TO mise à contribution dans le test (tests non pilotés seulement).

7.4.6.1 Classe de TO

Le nom du modèle de la classe des objets gérés.

7.4.6.2 Paramètre d'informations supplémentaires

Les noms des modèles de paramètre spécifiant les valeurs possibles de ce paramètre dans toute notification de résultat de test pour le TO.

7.4.7 Commentaires

Informations relatives à la catégorie de test. Par exemple: comportement gênant, relation entre les objets mis à contribution dans le test, corrélation des paramètres de demande et de réponse, enchaînements des réponses, ou sémantique supplémentaire des paramètres.

8 Définitions génériques

8.1 Types d'attributs génériques

Le présent article spécifie un ensemble d'attributs et/ou de paramètres génériques pouvant être inclus dans la définition des classes de TO, dans les notifications de résultats de test ou pouvant être utilisés comme paramètres d'invocation de test. Les modèles de ces définitions sont présentés dans l'Annexe A.

8.1.1 Objets associés

Quand il est présent, cet attribut ou paramètre identifie la ou les instances d'objet géré qui représentent une autre ressource mise à contribution dans le test. Des informations supplémentaires sur les objets associés peuvent également être indiquées. Dans un test de connectivité, par exemple, objet associé spécifie l'entité avec laquelle la ressource représentée par le MORT peut établir une connexion.

8.1.2 MORT

Quand il est présent, cet attribut ou paramètre identifie la ou les instances d'objet géré qui représentent la ressource en cours de test ou à tester. Ce paramètre est obligatoire dans toutes les notifications concernant l'invocation de test lorsqu'il figure également dans la demande de lancement de test.

8.1.3 Identificateur d'invocation de test

Cet attribut ou paramètre doit être présent; il identifie un test tel que spécifié au 7.3.1. Sa syntaxe comporte soit l'identificateur de l'objet ayant la fonctionnalité TARR qui a reçu la demande de test et un entier, soit le DN d'un TO comportant un test indépendant. Dans le cas d'une demande de test composé, la première syntaxe doit être utilisée. Pour les tests indépendants, la seconde peut être utilisée. C'est un attribut obligatoire pour tous les TO.

8.1.4 Identificateur d'objet de test

Cet attribut identifie un TO de manière unique. L'attribut de dénomination d'un TO est un identificateur d'objet de test, utilisé comme composant de RDN. Il peut être attribué soit par le responsable de test soit par l'exécuteur de test.

8.1.5 Résultat de test

Quand il est présent, cet attribut ou paramètre fournit une représentation normalisée d'un résultat de test. Ce paramètre peut apparaître dans une notification de résultats de test ou faire partie d'une réponse de test non piloté. Les conditions d'utilisation de cet attribut sont énoncées au 7.2.3.

Cet attribut peut prendre cinq valeurs:

- succès;
- échec;
- non concluant;
- hors des délais;
- terminaison prématurée.

ISO/CEI 10164-12 : 1994 (F)

Succès indique généralement que l'exécution du test s'est déroulée normalement et qu'aucune erreur n'a été détectée. Echec signifie que l'exécution du test s'est déroulée correctement, mais qu'un problème a été constaté. Non concluant indique qu'il est impossible de déterminer si le résultat est succès ou échec. Hors des délais indique que l'exécution du test a pris fin parce que le délai d'exécution a expiré ou à cause d'un conflit de planification. Terminaison prématurée indique que le test a pris fin prématurément, soit de manière spontanée, soit à la suite d'une demande.

La signification de chaque valeur et leurs conditions d'apparition peuvent faire l'objet d'une description plus détaillée pour une catégorie donnée.

8.1.6 Identificateur de session de test

Quand il est présent, cet attribut ou paramètre identifie une session de test telle que spécifiée au 7.3.2. Constitué d'un entier, il peut aussi comporter un identificateur d'objet ou un nom distinctif. Tous les exécuteurs de test doivent supporter tous les composants de l'identificateur de session de test.

8.1.7 Période de temporisation

Quand il est présent, cet attribut ou paramètre définit la période maximale de temps au cours de laquelle un test peut être lancé.

8.2 Blocs génériques

8.2.1 Bloc heure réelle de test

Ce bloc définit deux attributs permettant à l'exécuteur du test d'indiquer la période d'exécution du test. Il s'agit de l'heure réelle de lancement (heure absolue, la valeur par défaut est inconnue) et d'une heure réelle de fin (heure absolue, la valeur par défaut est inconnue).

8.2.2 Bloc instant d'initialisation

Ce bloc définit un attribut permettant d'indiquer l'heure à laquelle le TO passe dans l'état en cours d'initialisation. Il comporte l'attribut instant d'initialisation qui peut être une heure absolue ou une heure liée à celle à laquelle le TO passera dans l'état en cours de test.

8.2.3 Bloc fenêtre demandée

Ce bloc définit deux attributs permettant à un responsable de test de réguler la plage horaire dans laquelle un test doit être exécuté. Il comporte les attributs heure de début et heure de fin (les valeurs NULL signifient tout le temps). Si heure de début n'est pas fourni dans la demande de test, l'attribut aura pour valeur l'heure en cours. Si heure de début indique une heure plus tardive que heure de fin, le TO doit émettre une notification de conflit de planification.

8.2.4 Bloc étape de test

Ce bloc définit deux attributs identifiant une étape de test dans un des états de test définis. Il comporte un qualificateur d'étape de test et un ensemble facultatif d'informations concernant l'étape.

8.3 Notifications génériques

L'ensemble des notifications génériques, paramètres et sémantique définis dans la présente Recommandation | Norme internationale décrit les paramètres généraux suivants du service G-RAPPORT-ÉVÉNEMENT (M-EVENT-REPORT) défini dans la Rec. X.710 du CCITT | ISO/CEI 9595:

- type d'événement;
- informations sur l'événement;
- réponse à l'événement.

Toutes les notifications peuvent être entrées dans un journal de gestion des systèmes et la présente Recommandation | Norme internationale définit à cet effet une classe d'objets gérés. La Rec. X.721 du CCITT | ISO/CEI 10165-2 définit une classe d'objets génériques d'enregistrement de consignation d'événement, de laquelle sont dérivées toutes les entrées, les informations supplémentaires étant spécifiées par les paramètres informations sur l'événement et réponse à l'événement.

8.3.1 Type d'événement

Ce paramètre décrit la notification en indiquant s'il s'agit

- d'informations sur le test, le paramètre type de résultat de test étant défini à cet effet;
- d'une notification de conflit de planification, le paramètre type de compte rendu de conflit de planification étant défini à cet effet.

8.3.2 Informations sur l'événement

Les paramètres suivants, définis dans la présente Recommandation | Norme internationale, représentent les informations spécifiques sur la notification:

- heure réelle de début;
- heure réelle de fin;
- objets associés;
- heure de fin;
- MORT;
- heure de début;
- résultat de test;
- identificateur d'invocation de test;
- identificateur de session de test.

Les paramètres ci-dessous, définis dans la Rec. X.733 du CCITT | ISO/CEI 10164-4, sont également utilisés:

- informations supplémentaires;
- texte supplémentaire;
- notifications corrélées;
- attributs supervisés;
- identificateur de notification;
- actions correctives proposées.

8.4 Actions génériques

L'ensemble des paramètres d'actions génériques et de la sémantique définis par la présente Recommandation | Norme internationale décrit les paramètres généraux suivants du service G-ACTION (M-ACTION) défini dans la Rec. X.710 du CCITT | ISO/CEI 9595:

- type d'action;
- informations sur l'action;
- réponse à l'action.

8.4.1 Type d'action

Ce paramètre identifie l'action comme étant

- une demande de lancer un test piloté, ce type d'information sur la demande de test piloté étant défini dans ce but;
- une demande de lancer un test non piloté, ce type d'information sur la demande de test non piloté étant défini dans ce but;
- une demande de suspendre ou de reprendre un test, ce type d'information sur la suspension/reprise de test étant défini dans ce but;
- une demande de terminer un test, ce type d'information de terminaison de test étant défini dans ce but.

8.4.2 Informations sur l'action

Les paramètres suivants peuvent être véhiculés dans «informations sur l'action».

8.4.2.1 Type de demande de test piloté

Ce paramètre définit si la demande porte sur un seul test constitué de TO associés (test composé) ou sur des tests multiples où chaque test comporte un seul TO (test indépendant).

8.4.2.2 Tests indiqués

Ce paramètre indique les tests faisant l'objet d'une demande de pilotage. Il comporte soit un identificateur de session de test soit un ensemble d'identificateurs d'invocation de test.

8.4.2.3 Sélection suspension/reprise

Ce paramètre indique si le ou les tests spécifiés doivent être suspendus ou repris.

8.4.2.4 Information sur la catégorie de test

Ce paramètre facultatif indique les informations associées à une demande de test. La syntaxe et la sémantique de ces informations sont définies par un modèle de paramètre en GDMO, présenté dans le formulaire de catégorie de test. Ces informations peuvent inclure des données d'entrée nécessaires à l'exécution du test par l'exécuteur de test mais qui ne doivent pas nécessairement être dans les attributs de TO.

Ce paramètre peut être présent dans les demandes de tests pilotés, mais doit être présent dans les demandes de tests non pilotés.

8.4.2.5 Liste d'objets de test

Quand il est présent, ce paramètre spécifie le ou les TO devant être créés à la suite d'une demande de test piloté. Il indique la classe pour chaque TO et peut également fournir un nom d'instance de TO ainsi que des informations d'initialisation d'attributs de TO. L'ordre des informations de TO est important parce qu'il peut servir à établir la corrélation des informations de TO renvoyées dans une réponse, positive ou négative. Cet ordre est indépendant de l'ordre de création des TO qui relève d'une initiative locale.

8.4.2.6 MORT à tester

Quand il est présent, cet attribut ou paramètre identifie la ou les instances d'objet géré qui représentent les ressources à tester. Les MORT doivent être identifiés explicitement ou implicitement dans la demande de test. Le MORT peut être identifié implicitement comme étant l'objet disposant de la fonctionnalité TARR, ou un ou plusieurs MORT peuvent être fournis dans la liste d'attributs initiale des TO. Des MORT peuvent être spécifiés explicitement, soit dans une liste, soit par un mécanisme de définition du cadre de sélection et filtrage.

8.4.2.7 Autres informations

Les paramètres suivants, définis dans la présente Recommandation | Norme internationale, sont également utilisés:

- objets associés;
- période de temporisation;
- identificateur d'invocation de test;
- identificateur de session de test.

8.4.3 Réponse à l'action

Les paramètres suivants peuvent être véhiculés dans réponse à l'action.

8.4.3.1 Réponse de test indépendant

Quand il est présent, ce paramètre indique que tous les TO inclus dans la demande de test ont été instanciés et que le paramètre type de demande de test piloté indique indépendant. Ce paramètre renvoie des informations sur les TO instanciés y compris leur identificateur d'invocation de test et, si le comportement des TO le spécifie, les valeurs d'attributs des TO.

8.4.3.2 Réponse à une demande de test composé

Quand il est présent, ce paramètre indique que tous les TO mentionnés dans la demande de test ont été instanciés. Ce paramètre est utilisé lorsque le paramètre type de demande de test piloté indique composé. Ce paramètre renvoie l'identificateur d'invocation de test du test, les noms des TO instanciés et, si le comportement des TO le spécifie, les valeurs d'attributs des TO. Les informations sur les TO apparaissent dans le même ordre que dans la demande de test.

8.4.3.3 Liste de réponse d'objets de test

Quand il est présent, ce paramètre véhicule le nom des TO créés à la suite d'une demande de test piloté. Les informations sur les TO doivent figurer dans le même ordre que dans la demande de test pour permettre à l'exécuteur de test d'associer demande et réponse.

8.4.3.4 Réponse à une demande de test piloté

Ce paramètre doit figurer dans une réponse positive à une demande de test piloté. Si le paramètre type de demande de test piloté indique indépendant, les résultats sont fournis par le paramètre liste de réponse de test indépendant. Si ce n'est pas le cas, le paramètre réponse à une demande de test composé est utilisé.

8.4.3.5 Réponse à une demande de test non piloté

Ce paramètre doit figurer dans une réponse positive à une demande de test non piloté. Il fournit des informations sur les résultats d'un test non piloté. Il s'agit d'informations relatives aux MORT identifiés dans la demande de test.

8.4.3.6 Résultat d'une suspension/reprise de test

Quand il est présent, ce paramètre indique que l'objet géré disposant d'une fonctionnalité TARR a bien effectué la suspension ou la reprise de tous les tests demandés. L'identificateur d'invocation de test et l'état des TO sont spécifiés.

8.4.3.7 Autres informations

Les paramètres suivants, définis dans la présente Recommandation | Norme internationale, sont utilisés:

- objets associés;
- période de temporisation;
- MORT;
- identificateur d'invocation de test;
- état de test.

Les paramètres suivants, définis dans la Rec. X.733 du CCITT | ISO/CEI 10164-4, sont également utilisés:

- informations supplémentaires;
- texte supplémentaire;
- actions correctives proposées.

8.4.4 Erreurs spécifiques

Les paramètres suivants indiquent les informations transmises en cas d'erreur. Le paramètre échec d'une demande de test précise le type d'erreur.

8.4.4.1 Echec d'une demande de test

Ce paramètre figure dans la confirmation d'échec d'une demande de test (voir 7.2) dans le but de fournir des informations sur l'échec. Les différentes erreurs possibles à utiliser sont présentées dans les paragraphes suivants.

8.4.4.2 Erreur d'objet associé non disponible

Ce paramètre indique qu'un ou plusieurs des objets associés spécifiés ne sont pas dans un état approprié pour le test. La syntaxe du paramètre identifie le ou les objets associés qui ne sont pas dans l'état approprié.

8.4.4.3 Erreur d'invocation de test indépendant

Ce paramètre indique qu'un ou plusieurs des TO demandés n'ont pas pu être instanciés et que le paramètre type de demande de test piloté indiquait indépendant. Le paramètre retourne des informations sur la réussite ou l'échec de l'instanciation des TO. Les informations sur les TO figurent dans le même ordre que dans la demande de test.

8.4.4.4 Erreur d'opération de test non valable

Ce paramètre indique que l'opération de test demandée pour la classe d'objets gérés n'est pas valable. La syntaxe du paramètre donne l'identificateur d'objet de l'opération de test non valable.

8.4.4.5 Erreur d'information sur la catégorie de test de type incorrect

Ce paramètre indique que les informations sur la catégorie de test données dans la demande de test ne sont pas valables. La syntaxe du paramètre donne l'identificateur d'objet des informations non valables.

8.4.4.6 Erreur de MORT non disponible

Ce paramètre indique qu'un ou plusieurs des MORT ne sont pas dans un état où ils peuvent être testés. La syntaxe du paramètre identifie le ou les MORT qui ne sont pas dans un état approprié.

8.4.4.7 Erreur d'inexistence d'objet associé

Ce paramètre indique qu'un ou plusieurs des objets associés spécifiés n'existent pas. La syntaxe du paramètre identifie le ou les objets n'existant pas.

8.4.4.8 Erreur d'inexistence du MORT

Ce paramètre indique que l'objet ayant la fonctionnalité TARR n'a pas reconnu un ou plusieurs des MORT spécifiés dans le paramètre MORT à tester de la demande de test. La syntaxe du paramètre identifie le ou les MORT non reconnus.

8.4.4.9 Erreur d'inexistence de l'identificateur d'invocation de test

Ce paramètre indique qu'il n'existe pas de test ayant l'identificateur d'invocation de test spécifié. La syntaxe du paramètre identifie l'identificateur d'invocation de test.

8.4.4.10 Erreur d'inexistence de l'identificateur de session de test

Ce paramètre indique qu'il n'existe pas, dans la dépendance de l'objet daté de la fonctionnalité TARR, de test ayant l'identificateur de session de test spécifié. La syntaxe du paramètre identifie l'identificateur de session de test.

8.4.4.11 Erreur de TO composé

Ce paramètre indique que le test n'a pas pu être lancé car un ou plusieurs des TO demandés n'ont pas pu être créés et que le paramètre type de demande de test piloté indiquait composé. Le paramètre indique les TO qui ont pu être créés et ceux qui n'ont pas pu l'être. Les informations sur les TO figurent dans le même ordre que dans la demande de test.

8.4.4.12 Erreur de suspension/reprise de test

Quand il est présent, ce paramètre indique que l'objet géré doté de la fonctionnalité TARR n'a pas pu effectuer la suspension ou la reprise d'un ou de plusieurs tests spécifiés. L'identificateur d'invocation de test et l'état des TO sont indiqués aussi bien pour les tests pour lesquels l'opération a échoué que pour ceux pour lesquels elle a réussi.

8.4.4.13 Erreur de terminaison de test

Quand il est présent, ce paramètre indique que l'objet géré doté de la fonctionnalité TARR n'a pu terminer un ou plusieurs des tests spécifiés. L'identificateur d'invocation de test est indiqué aussi bien pour les tests qui n'ont pas été terminés que pour ceux qui ont été terminés.

8.5 Objets gérés

8.5.1 Enregistrement du conflit de planification

Enregistrement du conflit de planification est une classe d'objets gérés dérivée de la classe d'objets gérés enregistrement de consignation d'événement définie dans la Rec. X.721 du CCITT | ISO/CEI 10165-2. Elle représente des informations enregistrées dans des registres suite à la réception d'un compte rendu d'événement, le type d'événement étant défini au 9.10.

8.5.2 Exécuteur d'action de test

Exécuteur d'action de test est une classe d'objets gérés de support disposant de la fonctionnalité TARR, définie dans la présente Recommandation | Norme internationale. Cette classe peut être utilisée pour demander des tests pilotés ou non. La présente Recommandation | Norme internationale n'impose pas l'utilisation de cette classe d'objets gérés.

8.5.2.1 Essais non pilotés pris en charge

Cet attribut peut être utilisé pour identifier les tests non pilotés pris en charge par un objet géré doté de la fonctionnalité TARR. La valeur de l'attribut est l'ensemble d'identificateurs d'objet formé par les valeurs valides du champ information sur la catégorie de test de la demande de test non piloté destinée à cet objet.

8.5.2.2 Classes de TO prises en charge

Cet attribut permet d'identifier les classes de TO prises en charge par un objet géré doté de la fonctionnalité TARR.

8.5.2.3 Identificateur d'exécuteur d'action de test

Cet attribut permet d'identifier un objet d'exécuteur d'action de test.

8.5.3 Objet de test

Cette classe d'objets gérés est définie dans la présente Recommandation | Norme internationale pour permettre de définir des objets plus spécifiques par spécialisation.

8.5.4 Enregistrement de résultats de test

Enregistrement de résultats de test est une classe d'objets gérés dérivée de la classe d'objets gérés enregistrement de consignation d'événement, définie dans la Rec. X.721 du CCITT | ISO/CEI 10165-2. Elle représente des informations enregistrées dans des registres comme résultats de la réception d'un rapport d'événement lorsque le type d'événement est celui défini en 9.9.

NOTE – Les résultats de tests devant être consignés comprennent les résultats de tests pilotés et les résultats de tests non pilotés ainsi que des résultats de tests obtenus directement.

8.6 Définitions génériques importées

Les paramètres suivants, définis dans la Rec. X.733 du CCITT | ISO/CEI 10164-4 peuvent également être utilisés:

- texte additionnel;
- informations additionnelles;
- notifications corrélées;
- attributs supervisés;
- identificateur de notification;
- actions correctives proposées.

8.7 Conformité

Les définitions des classes d'objets gérés prennent en charge les fonctions définies dans la présente Recommandation | Norme internationale en incorporant la spécification des attributs, actions et notifications par référence aux modèles définis dans l'Annexe A et dans la Rec. X.721 du CCITT | ISO/CEI 10165-2. Le mécanisme de référence est défini dans la Rec. X.722 du CCITT | ISO/CEI 10165-4.

9 Définitions des services

9.1 Introduction

Le présent article définit les services permettant de demander le lancement, la suspension, la reprise et la terminaison des tests, ainsi que le compte rendu des résultats des tests et des conflits de planification.

9.2 Obtention des attributs de TO

Le service T-OBTENTION (PT-GET), défini dans la Rec. X.730 du CCITT | ISO/CEI 10164-1, peut être utilisé pour obtenir tout attribut lisible d'un TO.

9.3 Modification des attributs de TO

Le service T-MODIFICATION (PT-SET), défini dans la Rec. X.730 du CCITT | ISO/CEI 10164-1, peut être utilisé pour modifier tout attribut lisible d'un TO.

9.4 Abandon de tests pilotés

Le service T-SUPPRESSION (PT-DELETE), défini dans la Rec. X.730 du CCITT | ISO/CEI 10164-1, peut être utilisé pour permettre à un système ouvert de mettre fin à un test piloté dans un autre système ouvert. Lorsque le paramètre cadre de sélection existe, il est conseillé d'y regrouper tous les TO possédant le même identificateur d'invocation de test et de faire sélectionner les TO ayant le même identificateur d'invocation de test par le paramètre filtre.

Si une demande d'abandon entraîne la suppression de certains TO ayant le même identificateur d'invocation de test, le test devient indéterminé. Dans ce cas, la présente Recommandation | Norme internationale n'attache aucune sémantique à un échange ultérieur de protocole impliquant les TO restants.

9.5 Service de demande de test piloté

Le service de demande de test piloté permet à un gestionnaire (le responsable de test) de demander qu'un autre système ouvert (le système géré) lance un test piloté. Le Tableau 2 donne la liste des paramètres de ce service.

Outre les paramètres généraux du service G-ACTION (M-ACTION) définis dans la Rec. X.710 du CCITT | ISO/CEI 9595, le service de demande de test piloté utilise les paramètres définis à l'article 8.

NOTE – Les noms d'instances de TO ne doivent pas être fournis dans le paramètre liste d'objets de test si les paramètres cadre de sélection et filtre sont utilisés pour sélectionner des TARR multiples.

Le paramètre réponse à une demande de test piloté doit être présent dans toute réponse positive, sinon le paramètre erreurs doit être présent. En cas de réponse positive, le paramètre liste de réponses de tests indépendants ou le paramètre liste de réponses de tests composés doivent être présents et sont déterminés par la valeur du paramètre type de demande de test piloté. Dans les deux cas, les seules valeurs initiales que doit contenir le paramètre liste d'attributs de TO sont celles des attributs que le comportement de TO spécifie comme devant être renvoyés.

9.6 Service de demande de test non piloté

Ce service permet à un gestionnaire (responsable de test) de demander qu'un autre système ouvert (le système géré) lance et exécute un test non piloté. Le Tableau 3 donne la liste des paramètres de ce service.

Outre les paramètres généraux du service G-ACTION (M-ACTION) définis dans la Rec. X.710 du CCITT | ISO/CEI 9595, le service de demande de test non piloté utilise les paramètres définis à l'article 8.

Le paramètre liste de données de réponse de test doit être présent dans toute réponse positive, sinon le paramètre erreurs doit être présent.

Tableau 2 – Paramètres du service de demande de test piloté

Nom du paramètre	Demande/ indication	Réponse/ confirmation
Identificateur d'invocation	P	P
Identificateur lié	–	P
Mode	P	–
Classe de l'objet de base	P	–
Instance de l'objet de base	P	–
Cadre de sélection	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objets gérés	–	P
Commande d'accès	P	–
Synchronisation	P	–
Type de demande de test piloté	M	C(=)
Informations sur la demande de test piloté	M	–
Informations sur la catégorie de test	U	–
Type de demande de test piloté	M	–
Identificateur de session de test	U	–
MORT à tester	U	–
Objets associés	U	–
Période de temporisation	U	–
Liste d'objets de test	M	–
Classe de TO	M	–
Instance de TO	U	–
Instance de TO de référence	U	–
Liste initiale d'attributs	U	–
Date et heure	–	P
Réponse à une demande de test piloté	–	C
Réponse à une demande de test indépendant	–	C
Identificateur d'invocation de test	–	M
Instance de TO	–	C
Liste d'attributs de TO	–	C
Réponse à une demande de test composé	–	C
Identificateur d'invocation de test	–	M
Liste de réponses d'objets de test	–	M
Instance de TO	–	C
Attribut de TO	–	C
Erreurs	–	C

9.7 Service de suspension/reprise de test

Ce service permet à un gestionnaire (responsable de test) de demander qu'un autre système ouvert (le système géré) suspende ou reprenne un test ou une session de test. Ce service s'applique uniquement aux tests pilotés. Le Tableau 4 donne la liste des paramètres de ce service.

Outre les paramètres généraux du service G-ACTION (M-ACTION) définis dans la Rec. X.710 du CCITT | ISO/CEI 9595, le service de suspension/reprise de test utilise les paramètres définis à l'article 8.

Le paramètre résultat d'une suspension/reprise de test doit être présent dans toute réponse positive; sinon le paramètre erreurs doit être présent.

En réponse à une demande de suspension, le paramètre état de test doit indiquer l'état de test d'un TO avant la suspension. En réponse à une demande de reprise, le paramètre état de test doit indiquer l'état de test des TO après la reprise.

Tableau 3 – Paramètres du service de demande de test non piloté

Nom du paramètre	Demande/ indication	Réponse/ confirmation
Identificateur d'invocation	P	P
Identificateur lié	–	P
Mode	P	–
Classe de l'objet de base	P	–
Instance de l'objet de base	P	–
Cadre de sélection	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objets gérés	–	P
Commande d'accès	P	–
Synchronisation	P	–
Type de demande de test non piloté	M	C (=)
Informations de demande de test non piloté	M	–
Information sur la catégorie de test	M	–
Identificateur de session de test	U	C (=)
Période de temporisation	U	–
Objets associés	U	U
MORT à tester	U	–
Date et heure	–	P
Réponse à une demande de test non piloté	–	C
Résultat de test	–	C
MORT	–	U
Actions correctives proposées	–	U
Texte additionnel	–	U
Information additionnelle	–	U
Erreurs	–	C

Tableau 4 – Paramètres du service de suspension/reprise de test

Nom du paramètre	Demande/ indication	Réponse/ confirmation
Identificateur d'invocation	P	P
Identificateur lié	–	P
Mode	P	–
Classe de l'objet de base	P	–
Instance de l'objet de base	P	–
Cadre de sélection	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objets gérés	–	P
Commande d'accès	P	–
Synchronisation	P	–
Type de suspension/reprise de test	M	C(=)
Informations sur la suspension/reprise de test	M	–
Tests indiqués	M	–
Sélection suspension/reprise	M	–
Date et heure	–	P
Résultat d'une suspension/reprise de test	–	C
Identificateur d'invocation de test	–	M
Etat des objets de test	–	M
Instance de TO	–	C
Etat de test	–	M
Erreurs	–	C

9.8 Service de terminaison de test

Ce service permet à un gestionnaire (le responsable de test) de demander qu'un autre système ouvert (le système géré) termine un test ou une session de test. Ce service s'applique uniquement aux tests pilotés. Le Tableau 5 donne la liste des paramètres de ce service.

Outre les paramètres généraux du service G-ACTION (M-ACTION) définis dans la Rec. X.710 du CCITT | ISO/CEI 9595, le service de terminaison de test utilise les paramètres définis à l'article 8.

Le paramètre résultat de terminaison de test doit être présent dans une réponse positive; sinon, le paramètre erreurs doit être présent.

Tableau 5 – Paramètres du service de terminaison de test

Nom du paramètre	Demande/ indication	Réponse/ confirmation
Identificateur d'invocation	P	P
Identificateur lié	–	P
Mode	P	–
Classe de l'objet de base	P	–
Instance de l'objet de base	P	–
Cadre de sélection	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objets gérés	–	P
Commande d'accès	P	–
Synchronisation	P	–
Type de terminaison de test	M	C(=)
Informations de terminaison de test	M	–
Tests indiqués	M	–
Date et heure	–	P
Résultat de terminaison de test	–	C
Identificateur d'invocation de test	–	M
Erreurs	–	C

9.9 Service de résultat de test

Ce service permet à un système ouvert (le système géré) de rendre compte des résultats d'un test piloté. Le Tableau 6 donne la liste des paramètres de ce service.

Outre les paramètres généraux du service G-RAPPORT-ÉVÉNEMENT (M-EVENT-REPORT) définis dans la Rec. X.710 du CCITT | ISO/CEI 9595, le service de résultat de test utilise les paramètres définis à l'article 8.

Le paramètre identificateur d'invocation de test doit être présent si le TO a été créé comme membre d'un test composé, c'est-à-dire si type de demande de test piloté indiquait test composé quand le test a été lancé.

Le paramètre identificateur de session de test doit être présent si le responsable de test l'a fourni lors du lancement du test.

Le paramètre résultat de test doit être présent s'il s'agit du dernier compte rendu pour cette exécution du test sur ce TO.

Tableau 6 – Paramètres du service de résultat de test

Nom du paramètre	Demande/ indication	Réponse/ confirmation
Identificateur d'invocation	P	P
Mode	P	–
Classe d'objets gérés	P	–
Instance d'objets gérés	P	–
Type de résultat de test	M	C(=)
Date et heure de l'événement	P	–
Information sur le résultat du test		
Identificateur d'invocation de test	C	–
Identificateur de session de test	C	–
Résultat de test	C	–
MORT	U	–
Objets associés	U	–
Attributs supervisés	U	–
Actions correctives proposées	U	–
Texte additionnel	U	–
Informations additionnelles	U	–
Identificateur de notification	U	–
Notifications corrélées	U	–
Date et heure	–	P
Réponse à l'événement	–	–
Erreurs	–	C

9.10 Service de conflit de planification de test

Ce service permet à un système ouvert (le système géré) de rendre compte d'un conflit de planification. Le Tableau 7 donne la liste des paramètres de ce service.

Outre les paramètres généraux du service G-RAPPORT-ÉVÉNEMENT (M-EVENT-REPORT) définis dans la Rec. X.710 du CCITT | ISO/CEI 9595, le service de conflit de planification de test utilise les paramètres définis à l'article 8.

Le paramètre identificateur de session de test doit être présent si le responsable de test l'a fourni lors du lancement du test.

Tableau 7 – Paramètres du service de conflit de planification de test

Nom du paramètre	Demande/ indication	Réponse/ confirmation
Identificateur d'invocation	P	P
Mode	P	–
Classe d'objets gérés	P	–
Instance d'objets gérés	P	–
Type de compte rendu de conflit de planification	M	C(=)
Date et heure de l'événement	P	–
Information sur le conflit de planification		
Identificateur d'invocation de test	M	–
Identificateur de session de test	C	–
Heure de début	M	–
Heure de fin	M	–
Heure réelle de début	M	–
Heure réelle de fin	M	–
Texte additionnel	U	–
Informations additionnelles	U	–
Identificateur de notification	U	–
Notifications corrélées	U	–
Date et heure	–	P
Réponse à l'événement	–	–
Erreurs	–	C

10 Unités fonctionnelles de gestion de systèmes

La présente Recommandation | Norme internationale définit deux unités fonctionnelles pour la gestion des tests:

- a) une unité fonctionnelle de gestion des tests non pilotés;
- b) une unité fonctionnelle de gestion des tests pilotés.

L'unité fonctionnelle de gestion des tests non pilotés exige la prise en charge du service de demande de test non piloté. L'unité fonctionnelle de gestion des tests pilotés exige la prise en charge des services T-OBTENTION (PT-GET), T-MODIFICATION (PT-SET), T-SUPPRESSION (PT-DELETE) et des services de demande de test piloté, de résultat de test, de suspension/reprise de test, de terminaison de test et de conflit de planification.

11 Protocole

11.1 Eléments de procédure

11.1.1 Procédure de demande de test piloté

11.1.1.1 Rôle de gestionnaire

11.1.1.1.1 Invocation

Les procédures de demande de test piloté sont lancées par la primitive de demande de test piloté. Quand elle reçoit une primitive de demande de test piloté, la SMAPM doit construire une MAPDU et émettre une primitive de demande CMIS G-ACTION avec des paramètres dérivés de la primitive de demande de test piloté. Le mode confirmé doit être utilisé.

11.1.1.1.2 Réception de la réponse

Quand elle reçoit une primitive de confirmation CMIS G-ACTION contenant une MAPDU en réponse à une opération de demande de test piloté, la SMAPM doit envoyer une primitive de confirmation de demande de test piloté à l'utilisateur du service de demande de test piloté avec des paramètres dérivés de la primitive de confirmation CMIS G-ACTION, achevant ainsi la procédure de demande de test piloté.

NOTE – La SMAPM doit ignorer les erreurs dans la MAPDU. L'utilisateur du service de demande de test piloté peut ignorer ces erreurs ou couper l'association à cause de ces erreurs.

11.1.1.2 Rôle d'agent

11.1.1.2.1 Réception de la demande

Quand elle reçoit une primitive d'indication CMIS G-ACTION contenant une MAPDU demandant le service de demande de test piloté, la SMAPM doit, si la MAPDU est correcte, envoyer une primitive d'indication de demande de test piloté à l'utilisateur du service de demande de test piloté, avec des paramètres dérivés de la primitive d'indication CMIS G-ACTION. Autrement, la SMAPM doit construire une MAPDU appropriée indiquant l'erreur et émettre une primitive de réponse CMIS G-ACTION où figure le paramètre erreur.

11.1.1.2.2 Réponse

La SMAPM doit accepter une primitive de réponse à une demande de test piloté, construire une MAPDU confirmant l'opération et émettre une primitive de réponse CMIS G-ACTION avec des paramètres dérivés de la primitive de réponse à une demande de test piloté.

11.1.2 Procédure de demande de test non piloté

11.1.2.1 Rôle de gestionnaire

11.1.2.1.1 Invocation

Les procédures de demande de test non piloté sont lancées par la primitive de demande de test non piloté. Quand elle reçoit une primitive de demande de test non piloté, la SMAPM doit construire une MAPDU et émettre une primitive de demande CMIS G-ACTION avec des paramètres dérivés de la primitive de demande de test non piloté. Le mode confirmé doit être utilisé.

11.1.2.1.2 Réception de la réponse

Quand elle reçoit une primitive de confirmation CMIS G-ACTION contenant une MAPDU en réponse à une opération de demande de test non piloté, la SMAPM doit envoyer une primitive de confirmation de demande de test non piloté à l'utilisateur du service de demande de test non piloté avec des paramètres dérivés de la primitive de confirmation CMIS G-ACTION, achevant ainsi la procédure de demande de test non piloté.

NOTE – La SMAPM doit ignorer les erreurs dans la MAPDU. L'utilisateur du service de demande de test non piloté peut ignorer ces erreurs ou couper l'association à cause de ces erreurs.

11.1.2.2 Rôle d'agent

11.1.2.2.1 Réception de la demande

Quand elle reçoit une primitive d'indication CMIS G-ACTION contenant une MAPDU demandant le service de demande de test non piloté, la SMAPM doit, si la MAPDU est correcte, envoyer une primitive d'indication de demande de test non piloté à l'utilisateur du service de demande de test non piloté, avec des paramètres dérivés de la primitive d'indication CMIS G-ACTION. Autrement, la SMAPM doit construire une MAPDU appropriée indiquant l'erreur et émettre une primitive de réponse CMIS G-ACTION où figure le paramètre erreur.

11.1.2.2.2 Réponse

La SMAPM doit accepter la primitive de réponse à une demande de test non piloté, construire une MAPDU confirmant l'opération et émettre une primitive de réponse CMIS G-ACTION avec des paramètres dérivés de la primitive de réponse à une demande de test non piloté.

11.1.3 Procédure de suspension/reprise de test

11.1.3.1 Rôle de gestionnaire

11.1.3.1.1 Invocation

Les procédures de suspension/reprise de test sont lancées par la primitive de demande de suspension/reprise de test. Quand elle reçoit une primitive de demande de suspension/reprise de test, la SMAPM doit construire une MAPDU et émettre une primitive de demande CMIS G-ACTION avec des paramètres dérivés de la primitive de demande de suspension/reprise de test. Le mode confirmé doit être utilisé.

11.1.3.1.2 Réception de la réponse

Quand elle reçoit une primitive de confirmation CMIS G-ACTION contenant une MAPDU en réponse à une opération de suspension/reprise de test, la SMAPM doit envoyer une primitive de confirmation de suspension/reprise de test à l'utilisateur du service de suspension/reprise de test, avec des paramètres dérivés de la primitive de confirmation CMIS G-ACTION, achevant ainsi la procédure de suspension/reprise de test.

NOTE – La SMAPM doit ignorer les erreurs dans la MAPDU. L'utilisateur du service de suspension/reprise de test peut ignorer ces erreurs ou couper l'association à cause de ces erreurs.

11.1.3.2 Rôle d'agent

11.1.3.2.1 Réception de la demande

Quand elle reçoit une primitive d'indication CMIS G-ACTION contenant une MAPDU demandant le service de suspension/reprise de test, la SMAPM doit, si la MAPDU est correcte, envoyer une primitive d'indication de suspension/reprise de test à l'utilisateur du service de suspension/reprise de test, avec des paramètres dérivés de la primitive d'indication CMIS G-ACTION. Autrement, la SMAPM doit construire une MAPDU appropriée indiquant l'erreur et émettre une primitive de réponse CMIS G-ACTION où figure le paramètre erreur.

11.1.3.2.2 Réponse

La SMAPM doit accepter une primitive de réponse à une demande de suspension/reprise de test, construire une MAPDU confirmant l'opération et émettre une primitive de réponse CMIS G-ACTION avec des paramètres dérivés de la primitive de réponse de suspension/reprise de test.

11.1.4 Procédure de terminaison de test

11.1.4.1 Rôle de gestionnaire

11.1.4.1.1 Invocation

Les procédures de terminaison de test sont lancées par la primitive de demande de terminaison de test. Quand elle reçoit une primitive de demande de terminaison de test, la SMAPM doit construire une MAPDU et émettre une primitive de demande CMIS G-ACTION avec des paramètres dérivés de la primitive de demande de terminaison de test. Le mode confirmé doit être utilisé.

11.1.4.1.2 Réception de la réponse

Quand elle reçoit une primitive de confirmation CMIS G-ACTION contenant une MAPDU en réponse à une opération de terminaison de test, la SMAPM doit envoyer une primitive de confirmation de terminaison de test à l'utilisateur du service de terminaison de test, avec des paramètres dérivés de la primitive de confirmation CMIS G-ACTION, achevant ainsi la procédure de terminaison de test.

NOTE – La SMAPM doit ignorer les erreurs dans la MAPDU. L'utilisateur du service de terminaison de test peut ignorer ces erreurs ou couper l'association à cause de ces erreurs.

11.1.4.2 Rôle d'agent

11.1.4.2.1 Réception de la demande

Quand elle reçoit une primitive d'indication CMIS G-ACTION contenant une MAPDU demandant le service de terminaison de test, la SMAPM doit, si la MAPDU est correcte, envoyer une primitive d'indication de terminaison de test à l'utilisateur du service de terminaison de test, avec des paramètres dérivés de la primitive d'indication CMIS G-ACTION. Autrement, la SMAPM doit construire une MAPDU indiquant l'erreur et émettre une primitive de réponse CMIS G-ACTION où figure le paramètre erreur.

11.1.4.2.2 Réponse

La SMAPM doit accepter une primitive de réponse à une demande de terminaison de test, construire une MAPDU confirmant l'opération et émettre une primitive de réponse CMIS G-ACTION avec des paramètres dérivés de la primitive de réponse à la demande de terminaison de test.

11.1.5 Procédure de résultat de test

11.1.5.1 Rôle d'agent

11.1.5.1.1 Invocation

Les procédures de résultat de test sont lancées par la primitive de demande de résultat de test. Quand elle reçoit une primitive de demande de résultat de test, la SMAPM doit construire une MAPDU et émettre une primitive de demande CMIS G-RAPPORT-ÉVÉNEMENT avec des paramètres dérivés de la primitive de demande de résultat de test. En mode non confirmé, la procédure décrite en 11.1.5.1.2 ne s'applique pas.

11.1.5.1.2 Réception de la réponse

Quand elle reçoit une primitive de confirmation CMIS G-RAPPORT-ÉVÉNEMENT contenant une MAPDU en réponse à une notification de résultat de test, la SMAPM doit envoyer une primitive de confirmation de résultat de test à l'utilisateur du service de résultat de test, avec des paramètres dérivés de la primitive de confirmation CMIS G-RAPPORT-ÉVÉNEMENT, achevant ainsi la procédure de résultat de test.

NOTE – La SMAPM doit ignorer les erreurs dans la MAPDU. L'utilisateur du service de résultat de test peut ignorer ces erreurs ou couper l'association à cause de ces erreurs.

11.1.5.2 Rôle de gestionnaire

11.1.5.2.1 Réception de la demande

Quand elle reçoit une primitive d'indication CMIS G-RAPPORT-ÉVÉNEMENT contenant une MAPDU demandant le service de résultat de test, la SMAPM doit, si la MAPDU est correcte, envoyer une primitive d'indication de résultat de test à l'utilisateur du service de résultat de test, avec des paramètres dérivés de la primitive d'indication CMIS G-RAPPORT-ÉVÉNEMENT. Autrement, la SMAPM doit, en mode confirmé, construire une MAPDU appropriée indiquant l'erreur et envoyer une primitive de réponse CMIS G-RAPPORT-ÉVÉNEMENT où figure le paramètre erreur. En mode non confirmé, la procédure décrite en 11.1.5.2.2 ne s'applique pas.

11.1.5.2.2 Réponse

En mode confirmé, la SMAPM doit accepter une primitive de réponse de résultat de test, construire une MAPDU confirmant la notification et émettre une primitive de réponse CMIS G-RAPPORT-ÉVÉNEMENT avec des paramètres dérivés de la primitive de réponse de résultat de test.

11.1.6 Procédure de conflit de planification

11.1.6.1 Rôle d'agent

11.1.6.1.1 Invocation

Les procédures de conflit de planification sont lancées par la primitive de demande de conflit de planification. Quand elle reçoit une primitive de demande de conflit de planification, la SMAPM doit construire une MAPDU et émettre une primitive de demande CMIS G-RAPPORT-ÉVÉNEMENT avec des paramètres dérivés de la primitive de demande de conflit de planification. En mode non confirmé, la procédure décrite en 11.1.6.1.2 ne s'applique pas.

11.1.6.1.2 Réception de la réponse

Quand elle reçoit une primitive de confirmation CMIS G-RAPPORT-ÉVÉNEMENT contenant une MAPDU en réponse à une notification de conflit de planification, la SMAPM doit envoyer une primitive de confirmation de conflit de planification à l'utilisateur du service de conflit de planification avec des paramètres dérivés de la primitive de confirmation CMIS G-RAPPORT-ÉVÉNEMENT, achevant ainsi la procédure de conflit de planification.

NOTE – La SMAPM doit ignorer les erreurs dans la MAPDU. L'utilisateur du service de conflit de planification peut ignorer ces erreurs ou couper l'association à cause de ces erreurs.

11.1.6.2 Rôle de gestionnaire

11.1.6.2.1 Réception de la demande

Quand elle reçoit une primitive d'indication CMIS G-RAPPORT-ÉVÉNEMENT contenant une MAPDU demandant le service de conflit de planification, la SMAPM doit, si la MAPDU est correcte, envoyer une primitive d'indication de conflit de planification à l'utilisateur du service de conflit de planification, avec des paramètres dérivés de la primitive d'indication CMIS G-RAPPORT-ÉVÉNEMENT. Autrement la SMAPM doit, en mode confirmé, construire une MAPDU appropriée indiquant l'erreur et émettre une primitive de réponse CMIS G-RAPPORT-ÉVÉNEMENT où figure le paramètre erreur. En mode non confirmé, la procédure décrite en 11.1.6.2.2 ne s'applique pas.

11.1.6.2.2 Réponse

En mode confirmé, la SMAPM doit accepter une primitive de réponse de conflit de planification, construire une MAPDU confirmant la notification, et émettre une primitive de réponse CMIS G-RAPPORT-ÉVÉNEMENT avec des paramètres dérivés de la primitive de réponse de conflit de planification.

11.2 Syntaxe abstraite

11.2.1 Objets

La présente Recommandation | Norme internationale fait référence aux objets de support suivants dont la notation de syntaxe abstraite n° 1 est spécifiée dans l'Annexe A:

- a) schedulingConflictRecord;
- b) testActionPerformer;
- c) testObject;
- d) testResultsRecord.

11.2.2 Blocs

La présente Recommandation | Norme internationale fait référence aux blocs suivants dont la notation de syntaxe abstraite n° 1 est définie dans l'Annexe A:

- a) actualTestTimePackage;
- b) associatedObjectsPackage;
- c) controlledTestRequestPackage;
- d) initializingTimePackage;
- e) mORTsPackage;
- f) requestedWindowPackage;
- g) supportedTOClassesPackage;

- h) supportedUncontrolledTestsPackage;
- i) testActionPerformerPackage;
- j) testInvocationIdPackage;
- k) testObjectPackage;
- l) testOutcomePackage;
- m) testResultPackage;
- n) testSessionPackage;
- o) testStepsPackage;
- p) testSuspendResumePackage;
- q) testTerminatePackage;
- r) tOControlStatusPackage;
- s) uncontrolledTestRequestPackage.

11.2.3 Attributs

La présente Recommandation | Norme internationale fait référence aux attributs de gestion spécifiques suivants dont la syntaxe abstraite est spécifiée dans l'Annexe A:

- a) actualStartTime;
- b) actualStopTime;
- c) associatedObjects;
- d) endTime;
- e) initializingTime;
- f) mORTS;
- g) supportedTOClasses;
- h) supportedUncontrolledTests;
- i) testActionPerformerId;
- j) testInvocationId;
- k) testObjectId;
- l) testOutcome;
- m) testSessionId;
- n) testStep;
- o) testStepQualifier;
- p) timeoutPeriod.

11.2.4 Notifications

La présente Recommandation | Norme internationale fait référence aux types de notification spécifiques suivants dont la syntaxe abstraite est spécifiée dans l'Annexe A:

- a) schedulingConflictNotification;
- b) testResultNotification.

11.2.5 Actions

La présente Recommandation | Norme internationale fait référence aux types d'action spécifiques suivants dont la syntaxe abstraite est spécifiée dans l'Annexe A:

- a) testRequestControlledAction;
- b) testRequestUncontrolledAction;
- c) testSuspendResumeAction;
- d) testTerminateAction.

11.3 Négociation des unités fonctionnelles

La présente Recommandation | Norme internationale attribue l'identificateur d'objet suivant:

{ joint-iso-ccitt ms(9) function(2) part12(12) functionalUnitPackage(1) }

comme valeur de type ASN.1 FunctionalUnitPackageId défini dans la Rec. X.701 du CCITT | ISO/CEI 10040 à utiliser pour négocier les unités fonctionnelles suivantes:

- 0 unité fonctionnelle de gestion de test piloté;
- 1 unité fonctionnelle de gestion de test non piloté;

où le nombre désigne la position binaire affectée à l'unité fonctionnelle et où le nom désigne l'unité fonctionnelle définie à l'article 10.

Dans le cadre de la gestion des systèmes, le mécanisme de négociation des unités fonctionnelles de gestion de test est décrit dans la Rec. X.701 du CCITT | ISO/CEI 10040.

NOTE – Le besoin de négocier des unités fonctionnelles est spécifié par le contexte de l'application.

12 Relations avec d'autres fonctions

La notification de résultat de test définie dans la présente Recommandation | Norme internationale peut être enregistrée à l'aide des services décrits dans la Rec. X.735 du CCITT | ISO/CEI 10164-6.

Les MORT, les TO et les objets associés concernés par une invocation de test peuvent présenter des modifications d'état et avoir des valeurs de qualificateur d'état de test composé définies dans la Rec. X.731 du CCITT | ISO/CEI 10164-2.

Le lancement, la terminaison, la suspension et la reprise du service de RÉSULTAT DE TEST sont possibles en utilisant les services définis dans la Rec. X.734 du CCITT | ISO/CEI 10164-5, qui traitent des instances du discriminateur de retransmission d'événement.

Des TO peuvent être supprimés avec des services de gestion d'objets définis dans la Rec. X.730 du CCITT | ISO/CEI 10164-1. Les services de notification de création d'objet et de notification de suppression d'objet, définis dans la Rec. X.730 du CCITT | ISO/CEI 10164-1, peuvent être utilisés quand un TO est créé ou supprimé.

La Figure 6 représente ces relations.

13 Conformité

Il existe deux classes de conformité: une classe de conformité générale et une classe de conformité dépendante. Un système pour lequel est déclarée la mise en œuvre des éléments de procédure des services de gestion des systèmes définis dans la présente Recommandation | Norme internationale doit satisfaire aux conditions prescrites pour la classe de conformité générale ou de conformité dépendante, définies dans les paragraphes suivants. Le fournisseur doit indiquer la classe de conformité de sa version.

NOTE – L'utilisation des expressions «classe de conformité générale» et «classe de conformité dépendante» est à l'examen. La présente norme les utilise par souci de cohérence avec la Rec. X.710 du CCITT | ISO/CEI 10040 et les autres normes sous le titre général de *Technologie de l'information – Interconnexion de systèmes ouverts – Gestion des systèmes*. A l'issue de cet examen, il est prévu soit de clarifier, soit de corriger la présente clause de conformité ainsi que les dispositions connexes des autres Recommandations | Normes internationales.

13.1 Conditions de conformité générale

Un système pour lequel est déclarée la conformité générale doit remplir cette condition pour toutes les classes d'objets gérés qui importent les informations de gestion définies dans la présente Recommandation | Norme internationale.

NOTE – Cette condition s'applique à n'importe quelle sous-classe de classe d'objets de support de gestion définie dans la présente Recommandation | Norme internationale.

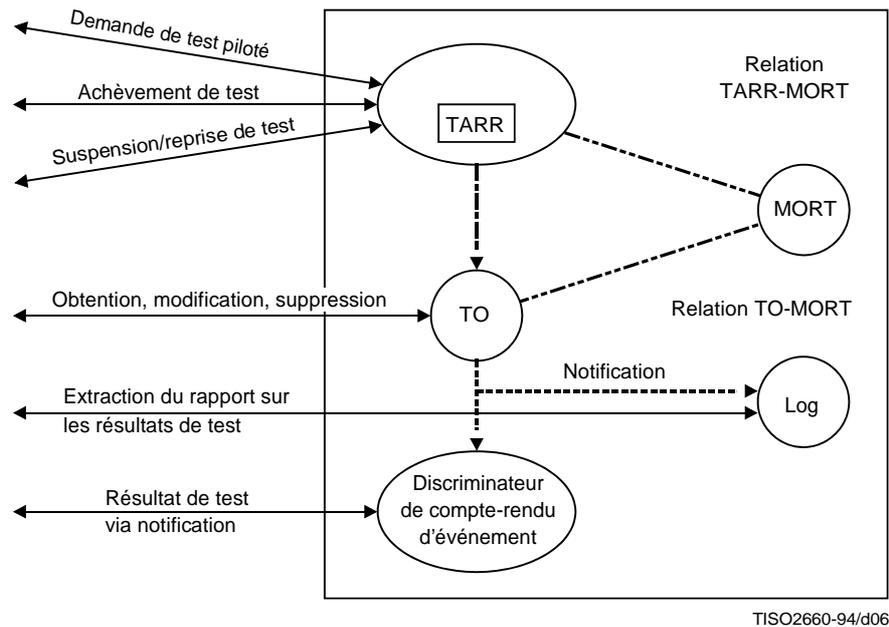


Figure 6 – Relations avec d'autres fonctions

13.1.1 Conformité statique

Le système doit:

- prendre en charge, dans les unités fonctionnelles de gestion des tests non pilotés et/ou pilotés, le rôle de gestionnaire ou d'agent, ou les deux;
- prendre en charge la syntaxe de transfert dérivée des règles de codage spécifiées dans la Rec. X.209 du CCITT | ISO/CEI 8825 et appelée {joint-iso-ccitt asn1(1) basic encoding(1)}, pour générer et interpréter les MAPDU, définies par les types de données abstraites mentionnés en 11.2.3, 11.2.4 et 11.2.5;
- dans le rôle d'agent, pour l'unité fonctionnelle de gestion des tests non pilotés, prendre en charge une instance d'une classe d'objets gérés avec un bloc demande de test non piloté; et
- dans le rôle d'agent, pour l'unité fonctionnelle de gestion des tests pilotés, prendre en charge une ou plusieurs instances de la classe d'objets de test ou l'une de ses sous-classes.

13.1.2 Conformité dynamique

Le système doit, dans le ou les rôles pour lesquels la conformité est déclarée:

- prendre en charge les éléments de procédure définis en 11.1;
- prendre en charge les éléments de procédure définis dans la Rec. X.730 du CCITT | ISO/CEI 10164-1 pour services T-OBTENTION (PT-GET), T-MODIFICATION (PT-SET) et T-SUPPRESSION (PT-DELETE).

13.2 Conditions de conformité dépendante

13.2.1 Conformité statique

Le système doit:

- a) prendre en charge la syntaxe de transfert dérivée des règles de codage spécifiées dans la Rec. X.209 du CCITT | ISO/CEI 8825 et appelée {joint-iso-ccitt asn1(1) basic encoding(1)}, pour générer et interpréter les MAPDU, définies par les types de données abstraites mentionnés en 11.2.3, 11.2.4 et 11.2.5, comme l'exige une spécification de référence;
- b) dans le rôle d'agent, pour l'unité fonctionnelle de gestion des tests non pilotés, prendre en charge une instance d'une classe d'objets gérés avec un bloc demande de test non piloté; ou
- c) dans le rôle d'agent, pour l'unité fonctionnelle de gestion des tests pilotés, prendre en charge une ou plusieurs instances de la classe d'objets de test ou de l'une de ses sous-classes.

13.2.2 Conformité dynamique

Le système doit prendre en charge les éléments de procédure définis ou mentionnés dans la présente Recommandation | Norme internationale, comme l'exige une spécification de référence.

13.3 Conformité aux définitions d'objets gérés de support

Les objets exécuteur d'action de test et les TO pris en charge par le système ouvert doivent respecter le comportement décrit à l'article 8 et la syntaxe énoncée dans l'Annexe A.

Annexe A

Définition d'informations de gestion

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

A.1 Generic Object Classes

A.1.1 Scheduling conflict record object class

schedulingConflictRecord MANAGED OBJECT CLASS

DERIVED FROM “Rec. X.721 | ISO/IEC 10165-2:1992”: eventLogRecord;
 CHARACTERIZED BY
 schedulingConflictRecordPackage PACKAGE
 BEHAVIOUR schedulingConflictRecordBehaviour BEHAVIOUR DEFINED AS
 “This managed object is used to represent information logged as a result of receiving a Scheduling conflict event report.”;
 ATTRIBUTES
 testInvocationId GET,
 “Rec. X.721 | ISO/IEC 10165-2:1992”: startTime GET,
 endTime GET,
 actualStartTime GET,
 actualStopTime GET;;;
 CONDITIONAL PACKAGES
 testSessionPackage PRESENT IF
 “a Test session identifier was present in the event report.”;
 REGISTERED AS { part12MObjectClass 1 };

A.1.2 Test action performer object class

testActionPerformer MANAGED OBJECT CLASS

DERIVED FROM “Rec. X.721 | ISO/IEC 10165-2:1992”: top;
 CHARACTERIZED BY testActionPerformerPackage;
 CONDITIONAL PACKAGES
 uncontrolledTestRequestPackage PRESENT IF
 “this functionality is supported. This package and/or the controlledTestRequest package shall be present.”,
 controlledTestRequestPackage PRESENT IF
 “this functionality is supported. This package and/or the uncontrolledTestRequest package shall be present.”,
 testSuspendResumePackage PRESENT IF
 “the controlledTestRequestPackage is present and this functionality is supported.”,
 testTerminatePackage PRESENT IF
 “the controlledTestRequestPackage is present and this functionality is supported.”,
 supportedTOClassesPackage PRESENT IF
 “the controlledTestRequestPackage is present.”,
 supportedUncontrolledTestsPackage PRESENT IF
 “the uncontrolledTestRequestPackage is present.”;
 REGISTERED AS { part12MObjectClass 2 };

A.1.3 TO object class

testObject MANAGED OBJECT CLASS

DERIVED FROM “Rec. X.721 | ISO/IEC 10165-2:1992”: top;
 CHARACTERIZED BY testObjectPackage;
 CONDITIONAL PACKAGES
 testOutcomePackage PRESENT IF
 “test results are to be retrieved by the Test conductor.”,
 testSessionPackage PRESENT IF
 “a Test session identifier was present in the test request.”,
 testResultPackage PRESENT IF
 “the test object is required to emit test result notifications.”,
 associatedObjectsPackage PRESENT IF
 “Associated objects were specified in the test request.”,

mORTsPackage PRESENT IF
 “MORT(s) are identified by the test object.”,
 tOControlStatusPackage PRESENT IF
 “the TO may exhibit the suspended state as specified in 7.3.3”,
 “Rec. X.721 | ISO/IEC 10165-2:1992”: availabilityStatusPackage PRESENT IF
 “scheduling is supported as defined in 7.2.2 or if the TO may exhibit the Idle state
 as specified in 7.3.3”,
 requestedWindowPackage PRESENT IF
 “scheduling is supported as defined in 7.2.2 and the test conductor may control the time at
 which the test is to be executed.”,
 actualTestTimePackage PRESENT IF
 “scheduling is supported as defined in 7.2.2 and the test performer may schedule the time at
 which the test is to be executed.”;
 REGISTERED AS { part12MObjectClass 3 };

A.1.4 Test results record object class

testResultsRecord MANAGED OBJECT CLASS

DERIVED FROM “Rec. X.721 | ISO/IEC 10165-2:1992”: eventLogRecord;
 CHARACTERIZED BY
 testResultsRecordPackage PACKAGE
 BEHAVIOUR testResultsRecordBehaviour BEHAVIOUR DEFINED AS
 “This managed object is used to represent information logged as a result of receiving a Test
 results event report.”;;;;
 CONDITIONAL PACKAGES
 testInvocationIdPackage PRESENT IF
 “a Test invocation identifier was present in the event report.”,
 testSessionPackage PRESENT IF
 “a Test session identifier was present in the event report.”,
 testOutcomePackage PRESENT IF
 “a Test outcome parameter was present in the event report.”,
 mORTsPackage PRESENT IF
 “a MORT(s) parameter was present in the event report.”,
 associatedObjectsPackage PRESENT IF
 “an Associated objects parameter was present in the event report.”,
 monitoredAttributesPackage PRESENT IF
 “a Monitored attributes parameter was present in the event report.”,
 proposedRepairActionsPackage PRESENT IF
 “a Proposed repair actions parameter was present in the event report.”;
 REGISTERED AS { part12MObjectClass 4 };

A.2 Package definitions

actualTestTimePackage PACKAGE

BEHAVIOUR actualTestTimePackageBehaviour BEHAVIOUR
 DEFINED AS “as specified in 7.2.2 and 8.2.1.”;;
 ATTRIBUTES actualStartTime DEFAULT VALUE Test-ASN1Module.defaultActualStartTime GET,
 actualStopTime DEFAULT VALUE Test-ASN1Module.defaultActualStopTime GET;
 REGISTERED AS { part12Package 1 };

associatedObjectsPackage PACKAGE

BEHAVIOUR associatedObjectsBehaviour BEHAVIOUR
 DEFINED AS “as specified in 8.1.1.”;;
 ATTRIBUTES associatedObjects GET;
 REGISTERED AS { part12Package 2 };

controlledTestRequestPackage PACKAGE

BEHAVIOUR controlledTestRequestBehaviour BEHAVIOUR
 DEFINED AS “indicates that the managed object supports the Test request controlled action and
 behaviour as specified in clauses 7 and 9.5.”;;
 ACTIONS testRequestControlledAction n o S u c h M O R T m O R T N o t A v a i l a b l e
 mistypedTestCategoryInformation noSuchAssociatedObject
 associatedObjectNotAvailable independentTestInvocationError
 relatedTOError;

REGISTERED AS { part12Package 3 };

initializingTimePackage PACKAGE

BEHAVIOUR initializingTimePackageBehaviour BEHAVIOUR
DEFINED AS “as specified in 7.2.2 and 8.2.2”;;

ATTRIBUTES initializingTime GET;

REGISTERED AS { part12Package 4 };

mORTsPackage PACKAGE

BEHAVIOUR mORTsBehaviour BEHAVIOUR
DEFINED AS “as defined in 8.1.2.”;;

ATTRIBUTES mORTs GET;

REGISTERED AS { part12Package 5 };

requestedWindowPackage PACKAGE

BEHAVIOUR requestedWindowBehaviour BEHAVIOUR
DEFINED AS “as specified in 7.2.2 and 8.2.3”;;

ATTRIBUTES “Rec. X.721 | ISO/IEC 10165-2:1992”: startTime DEFAULT VALUE DERIVED
RULES requestedWindowBehaviour
GET-REPLACE,
endTime DEFAULT VALUE Test-ASN1Module.defaultEndTime GET-REPLACE
REPLACE-WITH-DEFAULT;

REGISTERED AS { part12Package 6 };

supportedTOClassesPackage PACKAGE

BEHAVIOUR supportedTOClassesBehaviour BEHAVIOUR
DEFINED AS “as defined in 8.5.2.2.”;;

ATTRIBUTES supportedTOClasses GET;

REGISTERED AS { part12Package 7 };

supportedUncontrolledTestsPackage PACKAGE

BEHAVIOUR supportedUncontrolledTestsBehaviour BEHAVIOUR
DEFINED AS “as defined in 8.5.2.1.”;;

ATTRIBUTES supportedUncontrolledTests GET;

REGISTERED AS { part12Package 8 };

testActionPerformerPackage PACKAGE

BEHAVIOUR testActionPerformerBehaviour BEHAVIOUR
DEFINED AS “as specified in 8.5.2.”;;

ATTRIBUTES testActionPerformerId GET;

REGISTERED AS { part12Package 9 };

testInvocationIdPackage PACKAGE

BEHAVIOUR testInvocationIdPackageBehaviour BEHAVIOUR
DEFINED AS “as specified in 7.3.1.”;;

ATTRIBUTES testInvocationId GET;

REGISTERED AS { part12Package 10 };

testObjectPackage PACKAGE

BEHAVIOUR testObjectBehaviour BEHAVIOUR
DEFINED AS “as defined in clause 7. The behaviour of the Operational state and Procedural status
attributes in this context are specified in 7.3.3.”;;

ATTRIBUTES testObjectId GET,
testInvocationId GET,
“Rec. X.721 | ISO/IEC 10165-2:1992”: operationalState GET,
“Rec. X.721 | ISO/IEC 10165-2:1992”: proceduralStatus GET;

REGISTERED AS { part12Package 11 };

testOutcomePackage PACKAGE

BEHAVIOUR testOutcomeBehaviour BEHAVIOUR
DEFINED AS “as specified in 8.1.5.”;;

ATTRIBUTES testOutcome GET;

REGISTERED AS { part12Package 12 };

testResultPackage PACKAGE

NOTIFICATIONS testResultNotification;

REGISTERED AS { part12Package 13 };

testSessionPackage PACKAGE

BEHAVIOUR testSessionBehaviour BEHAVIOUR
DEFINED AS “as defined in 7.3.2 and 8.1.6.”;
ATTRIBUTES testSessionId GET;

REGISTERED AS { part12Package 14 };

testStepsPackage PACKAGE

BEHAVIOUR testStepsBehaviour BEHAVIOUR
DEFINED AS “as defined in 7.3.4 and 8.2.4.”;
ATTRIBUTES testStep GET,
testStepQualifier GET;

REGISTERED AS { part12Package 15 };

testSuspendResumePackage PACKAGE

BEHAVIOUR testSuspendResumePackageBehaviour BEHAVIOUR
DEFINED AS “as specified in 7.2.4 and 9.7.”;
ACTIONS

testSuspendResumeAction invalidTestOperation noSuchTestInvocationId
noSuchTestSessionId testSuspendResumeError;

REGISTERED AS { part12Package 16 };

testTerminatePackage PACKAGE

BEHAVIOUR testTerminatePackageBehaviour BEHAVIOUR
DEFINED AS “as specified in 7.2.5 and 9.8.”;
ACTIONS

testTerminateAction invalidTestOperation noSuchTestInvocationId
noSuchTestSessionId testTerminateError;

REGISTERED AS { part12Package 17 };

tOControlStatusPackage PACKAGE

BEHAVIOUR tOControlStatusBehaviour BEHAVIOUR
DEFINED AS “as specified in 7.3.3.”;
ATTRIBUTES “Rec. X.721 | ISO/IEC 10165-2:1992”:controlStatus GET;

REGISTERED AS { part12Package 18 };

uncontrolledTestRequestPackage PACKAGE

BEHAVIOUR uncontrolledTestRequestBehaviour BEHAVIOUR
DEFINED AS “indicates that the managed object supports the Test request uncontrolled action and
behaviour as specified in clause 7 and in 9.6.”;

ACTIONS testRequestUncontrolledAction n o S u c h M O R T m O R T N o t A v a i l a b l e
mistypedTestCategoryInformation
noSuchAssociatedObject
associatedObjectNotAvailable;

REGISTERED AS { part12Package 19 };

A.3 Attribute definitions

A.3.1 Actual start time

actualStartTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX Test-ASN1Module.ActualStartTime;
MATCHES FOR EQUALITY, ORDERING;

REGISTERED AS { part12AttributeId 1 };

A.3.2 Actual stop time

actualStopTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX Test-ASN1Module.ActualStopTime;
MATCHES FOR EQUALITY, ORDERING;

REGISTERED AS { part12AttributeId 2 };

A.3.3 Associated objects

associatedObjects ATTRIBUTE

WITH ATTRIBUTE SYNTAX Test-ASN1Module.AssociatedObjects;
MATCHES FOR EQUALITY;

REGISTERED AS { part12AttributeId 3 };

A.3.4 End time

endTime ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.EndTime;
 MATCHES FOR EQUALITY;
 REGISTERED AS { part12AttributeId 4 };

A.3.5 Initializing time

initializingTime ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.InitializingTime;
 MATCHES FOR EQUALITY;
 REGISTERED AS { part12AttributeId 5 };

A.3.6 MORT(s)

mORTs ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.MORTs;
 MATCHES FOR EQUALITY;
 REGISTERED AS { part12AttributeId 6 };

A.3.7 Supported TO classes

supportedTOClasses ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.SupportedTOClasses;
 MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
 REGISTERED AS { part12AttributeId 7 };

A.3.8 Supported uncontrolled Tests

supportedUncontrolledTests ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.SupportedUncontrolledTests;
 MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
 REGISTERED AS { part12AttributeId 8 };

A.3.9 Test action performer id

testActionPerformerId ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.TestActionPerformerId;
 MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
 REGISTERED AS { part12AttributeId 9 };

A.3.10 Test invocation identifier

testInvocationId ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.TestInvocationId;
 MATCHES FOR EQUALITY;
 REGISTERED AS { part12AttributeId 10 };

A.3.11 Test object id

testObjectId ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.TestObjectId
 MATCHES FOR EQUALITY;
 REGISTERED AS { part12AttributeId 11 };

A.3.12 Test outcome

testOutcome ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.TestOutcome;
 MATCHES FOR EQUALITY;
 REGISTERED AS { part12AttributeId 12 };

A.3.13 Test session identifier

testSessionId ATTRIBUTE
 WITH ATTRIBUTE SYNTAX Test-ASN1Module.TestSessionId;
 MATCHES FOR EQUALITY;
 REGISTERED AS { part12AttributeId 13 };

A.3.14 Test step

testStep ATTRIBUTE
WITH ATTRIBUTE SYNTAX Test-ASN1Module.TestStep;
MATCHES FOR EQUALITY, ORDERING;
REGISTERED AS { part12AttributeId 14 };

A.3.15 Test step qualifier

testStepQualifier ATTRIBUTE
WITH ATTRIBUTE SYNTAX Test-ASN1Module.TestStepQualifier;
MATCHES FOR EQUALITY;
REGISTERED AS { part12AttributeId 15 };

A.3.16 Timeout period

timeoutPeriod ATTRIBUTE
WITH ATTRIBUTE SYNTAX Test-ASN1Module.TimeoutPeriod;
MATCHES FOR EQUALITY;
REGISTERED AS { part12AttributeId 16 };

A.4 Action definitions

testRequestControlledAction ACTION
BEHAVIOUR testRequestControlledActionBehaviour;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Test-ASN1Module.TestRequestControlledInfo;
WITH REPLY SYNTAX Test-ASN1Module.TestRequestControlledResult;
REGISTERED AS { part12Action 1 };

testRequestControlledActionBehaviour BEHAVIOUR DEFINED AS
“When an action request as defined by 9.5 is received a controlled test shall be invoked as defined in clause 7. The information associated with the action request and specific behaviour of the test shall depend upon the Test category. The names of the TOs created for the test shall be returned in the response if no such names were supplied in the request.”;

testSuspendResumeAction ACTION
BEHAVIOUR testSuspendResumeBehaviour;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Test-ASN1Module.TestSuspendResumeInfo;
WITH REPLY SYNTAX Test-ASN1Module.TestSuspendResumeResult;
REGISTERED AS { part12Action 2 };

testSuspendResumeBehaviour BEHAVIOUR DEFINED AS
“When an action request as defined by 9.7 is received the test or tests specified by the request shall be suspended or resumed according to the Suspend/resume choice parameter. In the response to a suspend request the Test state parameter shall indicate the state of the TO before the test was suspended. In the response to a resume request the Test state parameter shall indicate the state at which the TO is being resumed.”;

testTerminateAction ACTION
BEHAVIOUR testTerminateBehaviour;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Test-ASN1Module.TestTerminateInfo;
WITH REPLY SYNTAX Test-ASN1Module.TestTerminateResult;
REGISTERED AS { part12Action 3 };

testTerminateBehaviour BEHAVIOUR DEFINED AS
“When an action request as defined by 9.8 is received the test or tests specified by the request shall be terminated. Result reports pertaining to associated TO(s) may be returned before the TO(s) are deleted as a side effect.”;

testRequestUncontrolledAction ACTION
BEHAVIOUR testRequestControlledActionBehaviour;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Test-ASN1Module.TestRequestUncontrolledInfo;
WITH REPLY SYNTAX Test-ASN1Module.TestRequestUncontrolledResult;
REGISTERED AS { part12Action 4 };

testRequestUncontrolledActionBehaviour BEHAVIOUR DEFINED AS
“When an action request as defined by 9.6 is received an uncontrolled test shall be invoked as defined in clause 7. The information associated with the action request and specific behaviour of the test shall depend upon the Test category.”;

A.5 Notification definitions

schedulingConflictNotification NOTIFICATION

BEHAVIOUR schedulingConflictBehaviour;
WITH INFORMATION SYNTAX Test-ASN1Module.SchedulingConflictInfo
AND ATTRIBUTE IDS

testInvocationId	testInvocationId,
testSessionId	testSessionId,
startTime	“Rec. X.721 ISO/IEC 10165-2:1992”: startTime,
endTime	endTime,
actualStartTime	actualStartTime,
actualStopTime	actualStopTime,
additionalText	“Rec. X.721 ISO/IEC 10165-2:1992”: additionalText,
additionalInformation	“Rec. X.721 ISO/IEC 10165-2:1992”: additionalInformation;

REGISTERED AS { part12Notification 1 };

schedulingConflictBehaviour BEHAVIOUR DEFINED AS

“This notification type is used to report scheduling conflicts as defined in 7.2.2 and 9.10.”;

testResultNotification NOTIFICATION

BEHAVIOUR testResultBehaviour;
WITH INFORMATION SYNTAX Test-ASN1Module.TestResultInfo
AND ATTRIBUTE IDS

testInvocationId	testInvocationId,
testSessionId	testSessionId,
testOutcome	testOutcome,
mORTs	mORTs,
associatedObjects	associatedObjects,
monitoredAttributes	“Rec. X.721 ISO/IEC 10165-2:1992”: monitoredAttributes,
proposedRepairActions	“Rec. X.721 ISO/IEC 10165-2:1992”: proposedRepairActions,
additionalText	“Rec. X.721 ISO/IEC 10165-2:1992”: additionalText,
additionalInformation	“Rec. X.721 ISO/IEC 10165-2:1992”: additionalInformation,
notificationIdentifier	“Rec. X.721 ISO/IEC 10165-2:1992”: notificationIdentifier,
correlatedNotifications	“Rec. X.721 ISO/IEC 10165-2:1992”: correlatedNotifications;

REGISTERED AS { part12Notification 2 };

testResultBehaviour BEHAVIOUR DEFINED AS

“This notification type is used to report test results as specified in 7.2.3 and 9.9.”;

A.6 Specific Error definitions

associatedObjectNotAvailable PARAMETER

CONTEXT SPECIFIC-ERROR ;
WITH SYNTAX Test-ASN1Module.AssociatedObjectNotAvailable;
BEHAVIOUR associatedObjectNotAvailableBehaviour BEHAVIOUR
DEFINED AS “as specified in 8.4.4.2”;

REGISTERED AS { part12Parameter 1 };

independentTestInvocationError PARAMETER

CONTEXT SPECIFIC-ERROR ;
WITH SYNTAX Test-ASN1Module.IndependentTestInvocationError;
BEHAVIOUR independentTestInvocationErrorBehaviour BEHAVIOUR
DEFINED AS “as defined in 8.4.4.3”;

REGISTERED AS { part12Parameter 2 };

invalidTestOperation PARAMETER

CONTEXT SPECIFIC-ERROR ;
WITH SYNTAX Test-ASN1Module.InvalidTestOperation;
BEHAVIOUR invalidTestOperationBehaviour BEHAVIOUR
DEFINED AS “as specified in 8.4.4.4”;

REGISTERED AS { part12Parameter 3 };

mistypedTestCategoryInformation PARAMETER

CONTEXT SPECIFIC-ERROR ;
WITH SYNTAX Test-ASN1Module.MistypedTestCategoryInformation;
BEHAVIOUR mistypedTestCategoryInformationBehaviour BEHAVIOUR
DEFINED AS “as defined in 8.4.4.5”;

REGISTERED AS { part12Parameter 4 };

mORTNotAvailable PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.MORTNotAvailable;
 BEHAVIOUR mORTNotAvailable BEHAVIOUR
 DEFINED AS “as specified in 8.4.4.6”;;

REGISTERED AS { part12Parameter 5 };

noSuchAssociatedObject PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.NoSuchAssociatedObject;
 BEHAVIOUR noSuchAssociatedObjectBehaviour BEHAVIOUR
 DEFINED AS “as defined in 8.4.4.7”;;

REGISTERED AS { part12Parameter 6 };

noSuchMORT PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.NoSuchMORT;
 BEHAVIOUR noSuchMORTBehaviour BEHAVIOUR
 DEFINED AS “as specified in 8.4.4.8”;;

REGISTERED AS { part12Parameter 7 };

noSuchTestInvocationId PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.NoSuchTestInvocationId;
 BEHAVIOUR noSuchTestInvocationIdBehaviour BEHAVIOUR
 DEFINED AS “as defined in 8.4.4.9”;;

REGISTERED AS { part12Parameter 8 };

noSuchTestSessionId PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.NoSuchTestSessionId;
 BEHAVIOUR noSuchTestSessionIdBehaviour BEHAVIOUR
 DEFINED AS “as specified in 8.4.4.10”;;

REGISTERED AS { part12Parameter 9 };

relatedTOError PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.RelatedTOError;
 BEHAVIOUR relatedTOErrorBehaviour BEHAVIOUR
 DEFINED AS “as defined in 8.4.4.11”;;

REGISTERED AS { part12Parameter 10 };

testSuspendResumeError PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.TestSuspendResumeError;
 BEHAVIOUR testSuspendResumeErrorBehaviour BEHAVIOUR
 DEFINED AS “as specified in 8.4.4.12”;;

REGISTERED AS { part12Parameter 11 };

testTerminateError PARAMETER

CONTEXT SPECIFIC-ERROR ;
 WITH SYNTAX Test-ASN1Module.TestTerminateError;
 BEHAVIOUR testTerminateErrorBehaviour BEHAVIOUR
 DEFINED AS “as defined in 8.4.4.13”;;

REGISTERED AS { part12Parameter 12 };

A.7 Abstract Syntax definitions

Test-ASN1Module { joint-iso-ccitt ms(9) function(2) part12(12) asn1Module(2) 0 }

DEFINITIONS IMPLICIT TAGS

```
 ::= BEGIN
 -- EXPORTS everything
 -- IMPORTS
    DistinguishedName
    FROM InformationFramework
    { joint-iso-ccitt ds(5) modules(1) informationFramework(1) }

    CMISFilter, Attribute, AttributeId, ObjectInstance, Scope
    FROM CMIP-1
    { joint-iso-ccitt ms(9) cmip(1) version1(1) protocol(3) }
```

StopTime, StartTime, CorrelatedNotifications, NotificationIdentifier, MonitoredAttributes,
 AdditionalInformation, AdditionalText, ProposedRepairActions, AttributeList, AvailabilityStatus,
 ControlStatus, ProceduralStatus, OperationalState
 FROM Attribute-ASN1Module
 { joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1 };

part12Action OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part12(12) action(9) }
 part12AttributeId OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part12(12) attribute(7) }
 part12MObjectClass OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part12(12) mObjectClass(3) }
 part12Notification OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part12(12) notification(10) }
 part12Package OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part12(12) package(4) }
 part12Parameter OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part12(12) parameter(5) }

A.7.1 Attribute definitions

ActualStartTime ::= CHOICE {
 unknown NULL,
 actualStart GeneralizedTime }

ActualStopTime ::= CHOICE {
 unknown NULL,
 actualStop GeneralizedTime }

AssociatedObjects ::= SET OF SEQUENCE {
 associatedObject ObjectInstance,
 associatedObjectInfo SEQUENCE {
 associatedObjectInfoId OBJECT IDENTIFIER,
 associatedObjectInform ANY DEFINED BY associatedObjectInfoId } OPTIONAL }

defaultActualStartTime ActualStartTime ::= unknown:NULL

-- The Actual start time remains unknown until the test performer has a real knowledge of the actual or expected time.

defaultEndTime EndTime ::= continual: NULL

EndTime ::= CHOICE {
 specific GeneralizedTime,
 relative Timespec,
 continual NULL }

InitializingTime ::= CHOICE {
 actualTime GeneralizedTime,
 relativeTime Timespec }

-- Timespec value forever [0] means NOW

InvocationId ::= SEQUENCE {
 tARRName DistinguishedName,
 testId INTEGER }

MORTs ::= SET OF ObjectInstance

SupportedTOClasses ::= SET OF OBJECT IDENTIFIER

SupportedUncontrolledTests ::= SET OF OBJECT IDENTIFIER

TestObjectId ::= INTEGER

TestStep ::= INTEGER

TestStepQualifier ::= GraphicString

TestOutcome ::= INTEGER {
 inconclusive(0),
 pass(1),
 fail(2),
 timed-out(3),
 premature-termination(4) }

Timespec ::= CHOICE {
 forever [22] NULL,
 hours [23] INTEGER,
 minutes [24] INTEGER,

seconds	[25]	INTEGER,
millisecs	[26]	INTEGER,
microsecs	[27]	INTEGER,
nanosecs	[28]	INTEGER }

TimeoutPeriod ::= Timespec

TestInvocationId ::= CHOICE {
 tOName [0] ObjectInstance,
 invocationId [1] InvocationId }

TestSessionId ::= SEQUENCE {
 localId INTEGER,
 globalRef [1] CHOICE { dnGlobRef DistinguishedName, oidGlobRef OBJECT IDENTIFIER }
 OPTIONAL }

A.7.2 Action Types

TestRequestControlledInfo ::= SEQUENCE {
 controlledTestRequestType [1] ControlledTestRequestType,
 testCategoryInformation [2] TestCategoryInformation OPTIONAL,
 testSessionId [3] TestSessionId OPTIONAL,
 toBeTestedMORTs [4] ToBeTestedMORTs OPTIONAL,
 associatedObjects [5] AssociatedObjects OPTIONAL,
 timeoutPeriod [6] TimeoutPeriod OPTIONAL,
 testObjectList [7] TestObjectList }

TestObjectList ::= SEQUENCE OF SEQUENCE {
 tOClass OBJECT IDENTIFIER,
 tOInstance [1] ObjectInstance OPTIONAL,
 referenceTOInstance [2] ObjectInstance OPTIONAL,
 initialAttributeList [3] AttributeList OPTIONAL }
 -- *initialAttributeList overrides any referenceTOInstance attribute values*

ControlledTestRequestType ::= ENUMERATED { independent(0), related(1) }

ToBeTestedMORTs ::= CHOICE {
 normalForm [29] SET OF ObjectInstance,
 scopedSet [30] SEQUENCE {
 base ObjectInstance,
 mORTsScope Scope DEFAULT baseObject,
 mORTsFilter CMISFilter DEFAULT "and:{ }" }

TestRequestControlledResponse ::= CHOICE {
 independentTestResponseList [0] SEQUENCE OF IndependentTestResponse,
 relatedTestResponse [1] RelatedTestResponse }

IndependentTestResponse ::= SEQUENCE {
 testInvocationId TestInvocationId,
 tOName TOName OPTIONAL,
 tOAttributeList AttributeList OPTIONAL }

RelatedTestResponse ::= SEQUENCE {
 testInvocationId TestInvocationId,
 testObjectResponseList SEQUENCE OF TestObjectResponse }

TestObjectResponse ::= SEQUENCE {
 tOName TOName,
 tOAttributeList AttributeList OPTIONAL }

TOName ::= CHOICE {
 conductorProvidedName NULL,
 performerProvidedName ObjectInstance }

TestRequestUncontrolledInfo ::= SEQUENCE {
 testCategoryInformation [1] TestCategoryInformation,
 testSessionId [2] TestSessionId OPTIONAL,
 timeoutPeriod [3] TimeoutPeriod OPTIONAL,
 associatedObjects [4] AssociatedObjects OPTIONAL,
 toBeTestedMORTs [5] ToBeTestedMORTs OPTIONAL }

TestRequestUncontrolledResult ::= SEQUENCE {
 testOutcome [0] **TestOutcome OPTIONAL,**
 mORTs [1] **SET OF ObjectInstance OPTIONAL,**
 proposedRepairActions [2] **ProposedRepairActions OPTIONAL,**
 additionalText [3] **AdditionalText OPTIONAL,**
 additionalInformation [4] **AdditionalInformation OPTIONAL }**

TestSuspendResumeInfo ::= SEQUENCE {
 indicatedTests **IndicatedTests,**
 suspendResumeChoice **SuspendResumeChoice }**

SuspendResumeChoice ::= ENUMERATED { suspend(0), resume(1) }

IndicatedTests ::= CHOICE {
 testSessionId [0] **TestSessionId,**
 testInvocationId [1] **SET OF TestInvocationId }**

TestSuspendResumeElement ::= SEQUENCE {
 testInvocationId [0] **TestInvocationId,**
 tOsStates [1] **SET OF TOsState }**

TestSuspendResumeResult ::= SET OF TestSuspendResumeElement

TOsState ::= SEQUENCE {
 tOInstance **ObjectInstance OPTIONAL,**
 -- tOInstance is optional if Test Invocation Id is a tOName --,
 testState **TestState }**

TestTerminateInfo ::= IndicatedTests

TestTerminateResult ::= SET OF TestInvocationId

TestState ::= SEQUENCE {
 operationalState **OperationalState,**
 proceduralStatus **ProceduralStatus,**
 controlStatus **ControlStatus OPTIONAL,**
 availabilityStatus **AvailabilityStatus OPTIONAL }**

A.7.3 Notification Types

TestResultInfo ::= SEQUENCE {
 testInvocationId [0] **TestInvocationId OPTIONAL,**
 testSessionId [1] **TestSessionId OPTIONAL,**
 testOutcome [2] **TestOutcome OPTIONAL,**
 mORTs [3] **MORTs OPTIONAL,**
 associatedObjects [4] **AssociatedObjects OPTIONAL,**
 monitoredAttributes [5] **MonitoredAttributes OPTIONAL,**
 proposedRepairActions [6] **ProposedRepairActions OPTIONAL,**
 additionalText [7] **AdditionalText OPTIONAL,**
 additionalInformation [8] **AdditionalInformation OPTIONAL,**
 notificationIdentifier [9] **NotificationIdentifier OPTIONAL,**
 correlatedNotifications [10] **CorrelatedNotifications OPTIONAL }**

SchedulingConflictInfo ::= SEQUENCE {
 testInvocationId **TestInvocationId,**
 testSessionId [1] **TestSessionId OPTIONAL,**
 startTime **StartTime,**
 endTime **EndTime,**
 actualStartTime **ActualStartTime,**
 actualStopTime **ActualStopTime,**
 additionalText [6] **AdditionalText OPTIONAL,**
 additionalInformation [7] **AdditionalInformation OPTIONAL }**

A.7.4 Parameter Types

NoSuchMORT ::= SET OF ObjectInstance

MORTNotAvailable ::= SET OF ObjectInstance

MistypedTestCategoryInformation ::= OBJECT IDENTIFIER

ISO/CEI 10164-12 : 1994 (F)

NoSuchTestInvocationId ::= TestInvocationId

NoSuchTestSessionId ::= TestSessionId

InvalidTestOperation ::= OBJECT IDENTIFIER

NoSuchAssociatedObject ::= SET OF ObjectInstance

AssociatedObjectNotAvailable ::= SET OF ObjectInstance

IndependentTestInvocationError ::= SEQUENCE OF CHOICE {
 testInstanceCreated **[0]** **IndependentTestResponse,**
 tONotCreated **TONotCreated }**

RelatedTOError ::= SEQUENCE OF CHOICE {
 ableToCreateTO **NULL,**
 unableToCreateTO **TONotCreated }**

TONotCreated ::= CHOICE {
 reason **ENUMERATED {**
 invalidTOclass(0),
 duplicateTOInstance(1),
 invalidMORTClass(2),
 invalidAssociatedObjectClass(3),
 mORTNotAvailable(4),
 associatedObjectNotAvailable(5) },
 inappropriateAttributeValue **AttributeList }**

TestSuspendResumeError ::= SET OF CHOICE {
 testSuspendResumeSuccess **[0]** **TestSuspendResumeElement,**
 testSuspendResumeFailure **[1]** **TestSuspendResumeElement }**

TestTerminateError ::= SET OF CHOICE {
 testTerminateSuccess **[0]** **TestInvocationId,**
 testTerminateFailure **[1]** **TestInvocationId }**

END

Annexe B

Exemples

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

La présente annexe contient des explications, des exemples et des séquences de protocole qui clarifient le modèle présenté dans l'article 7, ainsi que les façons d'utiliser le modèle pour satisfaire différents besoins. Elle est présentée à titre d'information.

B.1 Modèle pour des tests composés

Chaque invocation de test peut être exécutée séparément, mais il est possible de combiner les tests pour satisfaire un besoin de l'utilisateur. Cette combinaison s'appelle un test composé.

Pour lancer plusieurs tests simultanément, un responsable de test peut avoir besoin de plusieurs exécuteurs de test. De cette façon, il peut recueillir des informations en provenance de plusieurs systèmes différents, concernant l'ensemble particulier de circonstances examinées.

Pour exécuter le test demandé par le responsable de test, un exécuteur de test peut devoir mettre à contribution un ou plusieurs systèmes. Dans ce cas, un processus d'application du système contenant l'exécuteur de test peut agir comme responsable de test secondaire et demander aux exécuteurs de test (secondaires) d'autres systèmes de lancer des tests. A leur tour, ces opérateurs de test peuvent nécessiter la coopération d'autres systèmes, auquel cas un responsable de test appartenant au même système que l'exécuteur de test secondaire peut demander à d'autres exécuteurs de test de faire les tests, et ainsi de suite.

B.2 Exemples de modèle

Les figures et le texte ci-dessous décrivent la façon dont le modèle de test peut être utilisé.

B.2.1 Utilisation récurrente du modèle de test

Afin de satisfaire un besoin de l'utilisateur, il est possible, et même souhaitable, pour l'exécuteur de test de faire créer des instances supplémentaires du modèle. Ce concept est illustré à la Figure B.1.

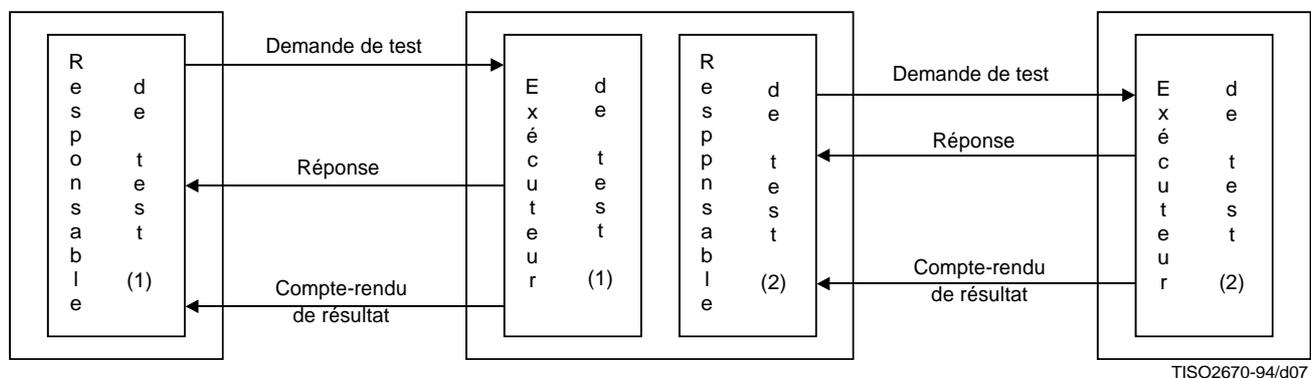


Figure B.1 – Utilisation récurrente du modèle de test

B.2.2 Utilisation parallèle du modèle de test

Afin de satisfaire un besoin de l'utilisateur, il est possible, et même souhaitable, pour un responsable de test d'exécuter plusieurs tests simultanément. Cette méthode est illustrée à la Figure B.2.

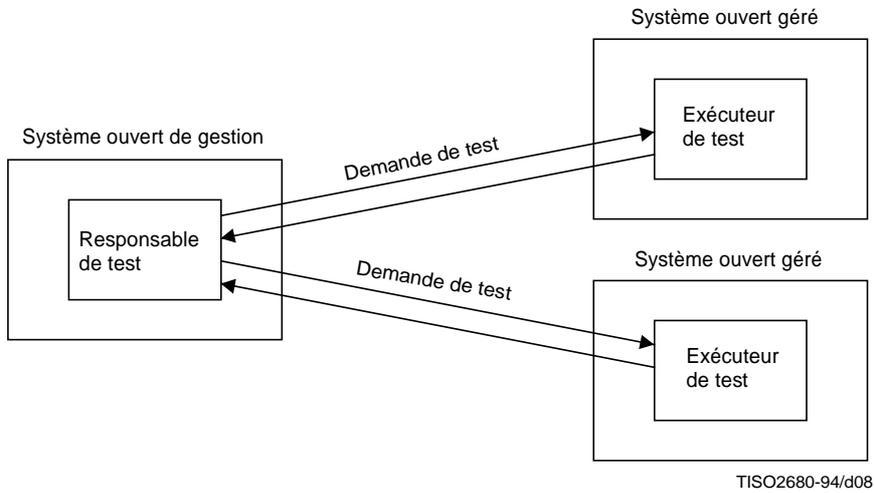


Figure B.2 – Utilisation parallèle du modèle de test

B.3 Séquences de protocole

Les figures et le texte ci-dessous décrivent les échanges de protocole de base imposés par le modèle, selon les différentes options utilisées: test non piloté ou piloté, compte rendu sollicité ou non sollicité, terminaison implicite ou explicite.

B.3.1 Terminaison explicite et compte rendu sollicité

La Figure B.3 montre un exemple de test piloté avec terminaison explicite et compte rendu sollicité. Le service T-OBTENTION (PT-GET) permet d'obtenir les résultats de test à n'importe quel état du test. Toutefois, si les résultats sont obtenus alors que le test est dans l'état en cours d'initialisation, les informations risquent de ne pas être fiables.

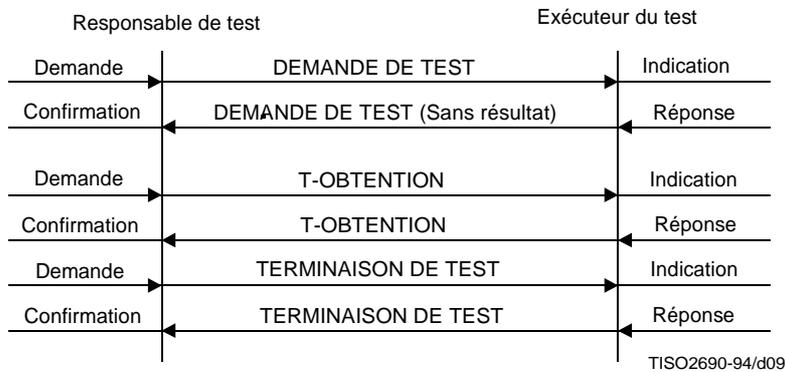


Figure B.3 – Terminaison explicite et compte rendu sollicité

B.3.2 Terminaison explicite et compte rendu non sollicité

La Figure B.4 montre un exemple de test piloté avec terminaison explicite et compte rendu non sollicité.

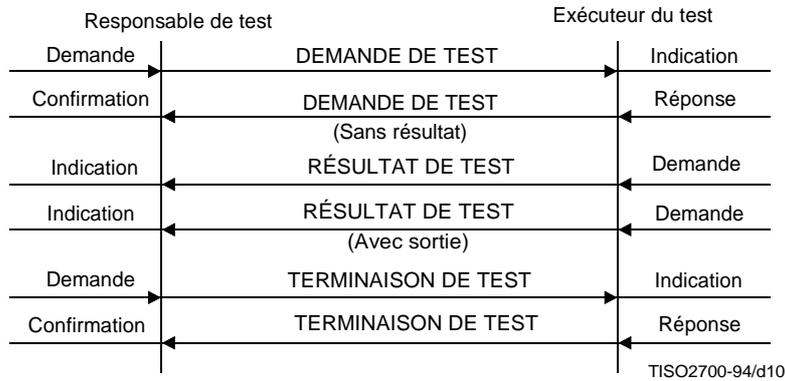


Figure B.4 – Terminaison explicite et compte rendu non sollicité

B.3.3 Terminaison implicite et compte rendu sollicité

La Figure B.5 montre un exemple de test non piloté avec terminaison implicite et compte rendu sollicité.

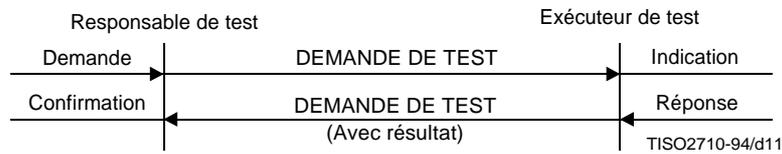


Figure B.5 – Terminaison implicite et compte rendu sollicité

B.3.4 Terminaison implicite et compte rendu non sollicité

La Figure B.6 montre un exemple de test piloté avec terminaison implicite et compte rendu non sollicité.

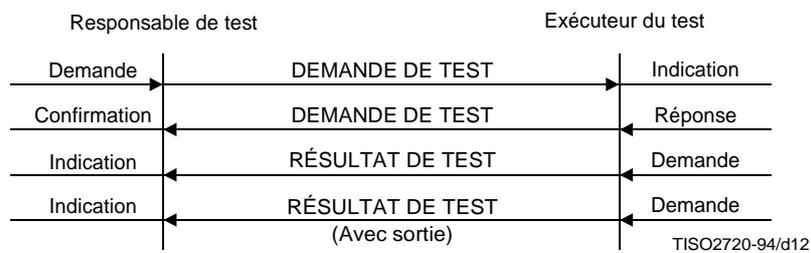


Figure B.6 – Terminaison implicite et compte rendu non sollicité

Abandon de tests pilotés, 22

Bloc étape de test, 16

Bloc fenêtre demandée, 16

Bloc heure réelle de test, 16

Bloc instant d'initialisation, 16

But de la catégorie de test, 14

Classe de TO, 15

Classes de TO prises en charge, 21

Compte rendu des résultats de test, 10

Conditions relatives à la fonctionnalité TARR, 14

Conditions relatives au TO, 14

Conditions relatives aux MORT, 14

Désactivé, 12

Echec d'une demande de test, 19

En cours d'achèvement, 12

En cours d'initialisation, 12

En cours de test, 12

Enregistrement de résultats de test, 21

Enregistrement du conflit de planification, 21

Erreur d'inexistence d'objet associé, 20

Erreur d'inexistence de l'identificateur d'invocation de test, 20

Erreur d'inexistence de l'identificateur de session de test, 20

Erreur d'inexistence du MORT, 20

Erreur d'information sur la catégorie de test de type incorrect, 20

Erreur d'invocation de test indépendant, 20

Erreur d'objet associé non disponible, 19

Erreur d'opération de test non valable, 20

Erreur de MORT non disponible, 20

Erreur de suspension/reprise de test, 20

Erreur de terminaison de test, 20

Erreur de TO composé, 20

Essais non pilotés pris en charge, 21

Etape de test, 14

Etats de test, 12

Exécuteur d'action de test, 21

Identificateur d'exécuteur d'action de test, 21

Identificateur d'invocation de test, 11, 15

Identificateur d'objet de test, 15

Identificateur de session de test, 12, 16

Inactif, 12
 Information sur la catégorie de test, 18
 Lancement d'un test, 9
 Liste d'objets de test, 18
 Liste de réponse d'objets de test, 19
 Modèle pour la fonction de gestion des tests, 7
 Modification des attributs de TO, 22
 MORT, 15
 MORT à tester, 18
 Nom de la catégorie de test, 14
 Non initialisé, 12
 Objet de test, 21
 Obtention des attributs de TO, 22
 Paramètre d'informations supplémentaires, 14
 Paramètre d'informations sur la catégorie de test, 14
 Pilotage de test, 14
 Planification des tests, 9
 Procédure de conflit de planification, 32
 Procédure de demande de test non piloté, 29
 Procédure de demande de test piloté, 29
 Procédure de résultat de test, 31
 Procédure de suspension/reprise de test, 30
 Procédure de terminaison de test, 30
 Réponse à une demande de test composé, 19
 Réponse à une demande de test non piloté, 19
 Réponse à une demande de test piloté, 19
 Réponse de test indépendant, 18
 Résultat d'une suspension/reprise de test, 19
 Résultat de test, 15
 Rôle d'agent, 31
 Rôle de gestionnaire, 31
 Scheduling conflict record object class, 37
 Sélection suspension/reprise, 18
 Service de conflit de planification de test, 27
 Service de demande de test non piloté, 22
 Service de demande de test piloté, 22
 Service de résultat de test, 26
 Service de suspension/reprise de test, 24
 Service de terminaison de test, 25
 Spécification d'informations de test, 14

ISO/CEI 10164-12 : 1994 (F)

Suspendu, 12

Suspension et reprise de test, 11

Terminaison de test, 11

Test action performer object class, 37

Test results record object class, 38

Tests indiqués, 18

Tests non pilotés, 7

Tests pilotés, 8

TO object class, 37

Type de demande de test piloté, 18

Type de service de demande de test, 14

Types d'attributs génériques, 15

Unités fonctionnelles de gestion de systèmes, 28