

Remplacée par une version plus récente



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

X.518

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(11/93)

**RÉSEAUX DE COMMUNICATION DE DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS
ANNUAIRE**

**TECHNOLOGIES DE L'INFORMATION –
INTERCONNEXION DES
SYSTÈMES OUVERTS –
L'ANNUAIRE: PROCÉDURES POUR LE
FONCTIONNEMENT RÉPARTI**

Recommandation UIT-T X.518
Remplacée par une version plus récente

(Antérieurement «Recommandation du CCITT»)

Remplacée par une version plus récente

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.518 de l'UIT-T a été approuvé le 16 novembre 1993. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 9594-4.

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.

Remplacée par une version plus récente

RECOMMANDATIONS UIT-T DE LA SÉRIE X RÉSEAUX POUR DONNÉES ET INTERCONNEXION DES SYSTÈMES OUVERTS

(Février 1994)

ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X

Domaine	Recommandations
RÉSEAUX PUBLICS POUR DONNÉES	
Services et services complémentaires	X.1-X.19
Interfaces	X.20-X.49
Transmission, signalisation et commutation	X.50-X.89
Aspects réseau	X.90-X.149
Maintenance	X.150-X.179
Dispositions administratives	X.180-X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200-X.209
Définition des services	X.210-X.219
Spécifications des protocoles en mode connexion	X.220-X.229
Spécifications des protocoles en mode sans connexion	X.230-X.239
Formulaires PICS	X.240-X.259
Identification des protocoles	X.260-X.269
Protocoles de sécurité	X.270-X.279
Objets gérés de couche	X.280-X.289
Test de conformité	X.290-X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Considérations générales	X.300-X.349
Systèmes mobiles de transmission de données	X.350-X.369
Gestion	X.370-X.399
SYSTÈMES DE MESSAGERIE	X.400-X.499
ANNUAIRE	X.500-X.599
RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES	
Réseautage	X.600-X.649
Dénomination, adressage et enregistrement	X.650-X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680-X.699
GESTION OSI	X.700-X.799
SÉCURITÉ	X.800-X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850-X.859
Traitement des transactions	X.860-X.879
Opérations distantes	X.880-X.899
TRAITEMENT OUVERT RÉPARTI	X.900-X.999

Remplacée par une version plus récente

TABLE DES MATIÈRES

	<i>Page</i>
Résumé.....	iv
Introduction.....	v
SECTION 1 – CONSIDÉRATIONS GÉNÉRALES.....	1
1 Domaine d'application.....	1
2 Références normatives.....	1
2.1 Recommandations Normes internationales identiques.....	1
2.2 Paires de Recommandations Normes internationales équivalentes par leur contenu technique.....	2
3 Définitions.....	2
3.1 Définitions relatives au modèle de référence OSI.....	2
3.2 Définitions de base relatives à l'Annuaire.....	2
3.3 Définitions relatives au modèle de l'Annuaire.....	2
3.4 Définition du modèle d'informations de DSA.....	3
3.5 Définitions relatives à la duplication dans l'Annuaire.....	3
3.6 Définitions relatives au fonctionnement réparti.....	3
4 Abréviations.....	5
5 Conventions.....	5
SECTION 2 – VUE D'ENSEMBLE.....	6
6 Vue d'ensemble.....	6
SECTION 3 – MODÈLES D'ANNUAIRE RÉPARTI.....	6
7 Modèle du système d'Annuaire réparti.....	6
8 Modèle des interactions entre les DSA.....	7
8.1 Décomposition d'une demande.....	8
8.2 Unichaînage.....	8
8.3 Multichaînage.....	8
8.4 Renvoi de référence.....	10
8.5 Détermination du mode.....	10
SECTION 4 – SERVICE ABSTRAIT DE DSA.....	11
9 Vue d'ensemble du service abstrait de DSA.....	11
10 Types d'information.....	11
10.1 Introduction.....	11
10.2 Types d'information définis ailleurs.....	12
10.3 Arguments de chaînage.....	12
10.4 Résultats du chaînage.....	14
10.5 Déroulement de l'opération.....	14
10.6 Information de trace.....	15
10.7 Type de référence.....	15
10.8 Information sur le point d'accès.....	15
10.9 Exclusions.....	16
10.10 Référence de continuation.....	17
11 Rattachement et détachement.....	18
11.1 DSA Bind (rattachement de DSA).....	18
11.2 DSA Unbind (Détachement de DSA).....	18

Remplacée par une version plus récente

Page

12	Opérations chaînées.....	18
12.1	Opérations chaînées.....	19
12.2	Opération ChainedAbandon.....	19
13	Erreurs chaînées.....	20
13.1	Introduction.....	20
13.2	Renvoi de référence de DSA.....	20
SECTION 5 – PROCÉDURES DE FONCTIONNEMENT RÉPARTI.....		20
14	Introduction.....	20
14.1	Domaine d'application et limites.....	20
14.2	Conformité.....	21
14.3	Modèle conceptuel.....	21
14.4	Fonctionnement individuel et coopératif des DSA.....	21
14.5	Accords de coopération entre DSA.....	21
15	Comportement de l'Annuaire réparti.....	22
15.1	Exécution coopérative des opérations.....	22
15.2	Phases de traitement d'une opération.....	22
15.3	Gestion des opérations réparties.....	23
15.4	Traitement des boucles.....	24
15.5	Autres considérations relatives au fonctionnement réparti.....	24
15.6	Authentification des opérations réparties.....	25
16	L'aiguilleur d'opérations.....	26
16.1	Principes généraux.....	26
16.2	Procédure de l'aiguilleur d'opérations.....	29
16.3	Vue d'ensemble des procédures.....	30
17	Validation de demande.....	32
17.1	Introduction.....	32
17.2	Paramètres de la procédure.....	32
17.3	Définition de la procédure.....	33
18	Résolution du nom.....	35
18.1	Introduction.....	35
18.2	Paramètres de la procédure Find DSE.....	35
18.3	Procédures.....	36
19	Evaluation de l'opération.....	43
19.1	Procédure de modification.....	43
19.2	Procédure d'interrogation à entrée unique.....	51
19.3	Procédure d'interrogation à entrées multiples.....	52
20	Procédures de référence de continuation.....	60
20.1	Stratégie de chaînage en présence de duplication miroir.....	62
20.2	Emission de sous-demandes chaînées vers un DSA distant.....	63
20.3	Paramètres des procédures.....	63
20.4	Définition des procédures.....	65
20.5	Procédure d'abandon.....	72
21	Procédure de fusionnement des résultats.....	72
22	Procédures d'authentification répartie.....	73
22.1	Authentification de l'expéditeur.....	73
22.2	Authentification des résultats.....	77
SECTION 6 – ADMINISTRATION DES CONNAISSANCES.....		77
23	Vue d'ensemble de l'administration des connaissances.....	77
23.1	Administration des références de connaissances.....	78
23.2	Demande de références croisées.....	79
23.3	Incohérences de connaissances.....	80

Remplacée par une version plus récente

	<i>Page</i>
24 Rattachements opérationnels hiérarchiques	81
24.1 Caractéristiques du type de rattachement opérationnel	81
24.2 Définition de la classe d'objets d'information de type operational-binding.....	83
24.3 Procédures de DSA pour la gestion des rattachements opérationnels hiérarchiques	84
24.4 Procédures pour les opérations.....	88
24.5 Utilisation des contextes d'application	88
25 Rattachement opérationnel hiérarchique non spécifique	88
25.1 Caractéristiques typiques du rattachement opérationnel	88
25.2 Définition de la classe d'objets d'information de rattachement opérationnel.....	89
25.3 Procédures relatives au DSA pour la gestion des rattachements opérationnels non spécifiques.....	90
25.4 Procédures applicables aux opérations.....	92
25.5 Utilisation des contextes d'application	92
Annexe A – ASN.1 pour les opérations réparties	93
Annexe B – Exemple de résolution répartie du nom	96
Annexe C – Utilisation répartie de l'authentification	98
C.1 Résumé	98
C.2 Authentification simple	98
C.3 Modèle d'authentification répartie.....	98
C.4 DUA à DSA	99
C.5 Transfert du DAP au DSP	99
C.6 Chaînage au travers de DSA intermédiaires.....	99
C.7 Authentification des résultats	99
Annexe D – Spécification des types de rattachement opérationnel hiérarchique et de rattachement opérationnel hiérarchique non spécifique.....	103
Annexe E – Exemple d'administration des connaissances.....	105
Annexe F – Amendements et correctifs.....	108

Remplacée par une version plus récente

Résumé

La présente Recommandation | Norme internationale spécifie les procédures grâce auxquelles les composantes réparties de l'Annuaire interfonctionnent pour fournir un service homogène aux utilisateurs.

Remplacée par une version plus récente

Introduction

La présente Recommandation | partie de Norme internationale a été élaborée de concert avec les autres Recommandations | Normes internationales, en vue de faciliter l'interconnexion des systèmes informatiques visant à assurer des services d'annuaire. L'ensemble de tous ces systèmes, avec les informations d'annuaire qu'ils contiennent, peut être considéré comme un tout intégré, appelé *Annuaire*. Les informations de l'Annuaire, désignées collectivement comme la base d'informations d'Annuaire (DIB), sont normalement utilisées pour faciliter la communication entre ou avec des objets tels que des entités d'application, des personnes, des terminaux et des listes de diffusion, ou à leurs propos.

L'Annuaire joue un rôle significatif dans l'interconnexion des systèmes ouverts (OSI) dont l'objectif est de permettre, au prix d'un minimum d'accords techniques en dehors des normes d'interconnexion proprement dites, d'interconnecter des équipements informatiques:

- de constructeurs différents;
- gérés de façon différentes;
- de niveaux de complexité différents;
- d'âges différents.

La présente Recommandation | Norme internationale spécifie les procédures d'interfonctionnement que les composants répartis de l'Annuaire mettent en œuvre pour fournir un service cohérent à ses utilisateurs.

Cette seconde édition révisé techniquement et améliore, mais ne remplace pas, la première édition de la présente Recommandation | Norme internationale. Les mises en œuvre peuvent encore prétendre à la conformité à la première édition.

Cette seconde édition spécifie la version 1 des protocoles et services de l'Annuaire. La première édition spécifie également la version 1. On a traité les différences entre les services et les protocoles définis dans les deux éditions en utilisant les règles d'extensibilité définies dans la présente version de la Rec. X.519 | ISO/CEI 9594-5.

L'Annexe A, qui fait partie intégrante de la présente Recommandation | Norme internationale, fournit le module de notation ASN.1 pour les opérations réparties de l'annuaire.

L'Annexe B, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit un exemple de résolution répartie du nom.

L'Annexe C, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit l'authentification dans un environnement d'opérations réparties.

L'Annexe D, qui fait partie intégrante de la présente Recommandation | Norme internationale, fournit les définitions des classes d'objets d'information en notation ASN.1 qui sont introduites dans la présente Spécification d'Annuaire.

L'Annexe E, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit la maintenance des connaissances.

L'Annexe F, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, énumère les amendements et relevés de défauts qui ont été incorporés pour former la présente édition de cette Recommandation | Norme internationale.

TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES OUVERTS – L'ANNUAIRE: PROCÉDURES POUR LE FONCTIONNEMENT RÉPARTI

SECTION 1 – CONSIDÉRATIONS GÉNÉRALES

1 Domaine d'application

La présente Recommandation | Norme internationale spécifie le comportement des agents de système d'Annuaire (DSA) (*directory system agents*) qui participent à la mise en œuvre répartie de l'Annuaire. Le comportement autorisé a été conçu pour assurer un service cohérent, compte tenu d'une large répartition de la base d'informations Annuaire (DIB) (*directory information base*) entre de nombreux DSA.

Bien qu'il puisse être construit à partir de tels systèmes, l'Annuaire n'est pas destiné à être un système de base de données à usage général. La fréquence des demandes est censée être considérablement plus élevée que celle des mises à jour.

2 Références normatives

Les Recommandations et les Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | partie de Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et 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 et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT-T tient à jour une liste des Recommandations UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.500 (1993) | ISO/CEI 9594-1:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Vue d'ensemble des concepts, modèles et services.*
- Recommandation UIT-T X.501 (1993) | ISO/CEI 9594-2:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Les modèles.*
- Recommandation UIT-T X.511 (1993) | ISO/CEI 9594-3:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Définition du service abstrait.*
- Recommandation UIT-T X.519 (1993) | ISO/CEI 9594-5:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Spécifications du protocole.*
- Recommandation UIT-T X.520 (1993) | ISO/CEI 9594-6:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Types d'attributs sélectionnés.*
- Recommandation UIT-T X.521 (1993) | ISO/CEI 9594-7:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Classes d'objets sélectionnés.*
- Recommandation UIT-T X.509 (1993) | ISO/CEI 9594-8:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Cadre d'authentification.*
- Recommandation UIT-T X.525 (1993) | ISO/CEI 9594-9:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Duplication.*

- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Spécification de la notation de base.*
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Spécification des objets informationnels.*
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Spécification des contraintes.*
- Recommandation UIT-T X.683 (1994) | ISO/CEI 8824-4:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Paramétrisation des spécifications de la notation de syntaxe abstraite n° 1.*
- Recommandation UIT-T X.880 (1994) | ISO/CEI 13712-1:1994, *Technologie de l'information – Opérations distantes: Concepts, modèle et notation.*
- Recommandation UIT-T X.881 (1994) | ISO/CEI 13712-2:1994, *Technologie de l'information – Opérations distantes: Réalisations OSI – Définition du service de l'élément de service d'opérations distantes.*

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

- Recommandation X.200 du CCITT (1988), *Modèle de référence de base pour l'interconnexion des systèmes ouverts pour les applications du CCITT.*
ISO 7498:1984/Corr.1:1988, *Systèmes de traitement de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base.*

3 Définitions

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

3.1 Définitions relatives au modèle de référence OSI

Le terme suivant est défini dans la Rec. X.200 du CCITT et dans ISO 7498:

- *titre d'entité d'application.*

3.2 Définitions de base relatives à l'Annuaire

Les termes suivants sont définis dans la Rec. UIT-T X.500 | ISO/CEI 9594-1:

- a) *(l')Annuaire,*
- b) *Base d'informations Annuaire.*

3.3 Définitions relatives au modèle de l'Annuaire

Les termes suivants sont définis dans la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- a) *point d'accès;*
- b) *alias;*
- c) *nom distinctif;*
- d) *arbre d'informations Annuaire;*
- e) *agent de système d'Annuaire;*
- f) *agent utilisateur d'Annuaire;*
- g) *nom distinctif relatif;*

9594-4 : 1995 (F)

3.4 Définition du modèle d'informations de DSA

Les termes suivants sont définis dans la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- a) *catégorie;*
- b) *d'utilisation commune;*
- c) *préfixe de contexte;*
- d) *référence croisée;*
- e) *fragment de base DIB;*
- f) *arbre des informations de DSA;*
- g) *entrée spécifique de DSA (DSE);*
- h) *type d'entrée DSE;*
- i) *référence immédiatement supérieure;*
- j) *informations de connaissance;*
- k) *catégorie de référence de connaissance;*
- l) *type de référence de connaissance;*
- m) *contexte de dénomination;*
- n) *connaissance non spécifique;*
- o) *référence subordonnée non spécifique;*
- p) *attribut opérationnel;*
- q) *chemin de référence;*
- r) *connaissance spécifique;*
- s) *référence subordonnée;*
- t) *référence supérieure.*

3.5 Définitions relatives à la duplication dans l'Annuaire

Les termes suivants définis dans la Rec. UIT-T X.525 | ISO/CEI 9594-9:

- a) *exhaustivité d'attribut;*
- b) *rattachement opérationnel de duplication miroir;*
- c) *exhaustivité de subordonnées;*
- d) *unité de duplication.*

3.6 Définitions relatives au fonctionnement réparti

Les termes suivants sont définis dans la présente Recommandation | Norme internationale:

3.6.1 entrée d'objet de base: Entrée d'objet ou alias qui est l'objectif d'une opération telle qu'il est émis par l'origine.

3.6.2 chaînage: Terme générique pouvant désigner l'unichainage et le multichainage.

3.6.3 information du préfixe de contexte: Information opérationnelle et d'utilisateur fournie par le DSA supérieur au DSA subordonné dans un RHOB concernant les nœuds d'arbre DIT supérieurs au préfixe de contexte subordonné.

3.6.4 résolution répartie du nom: Processus par lequel la résolution du nom est effectuée dans plus d'un DSA.

3.6.5 erreur: Information envoyée par l'exécuteur au demandeur et acheminant un résultat négatif concernant une demande précédemment reçue.

3.6.6 erreur permanente: Erreur définie qui indique que l'opération ne peut pas être accomplie actuellement sans intervention extérieure.

3.6.7 rattachement opérationnel hiérarchique (HOB): Relation entre deux DSA maîtres détenant des contextes de dénomination, dont l'un est immédiatement subordonné à l'autre, où le DSA supérieur détient une référence subordonnée au DSA subordonné.

3.6.8 opérations de modification: Opérations de type Directory Modify (modification d'Annuaire), comme Modify Entry (modifier entrée), Add Entry (ajouter entrée), Remove Entry (supprimer entrée) et Modify DN (modifier nom distinctif).

3.6.9 multichânage: Mode d'interaction dans lequel un DSA traitant une demande lui-même, envoie des demandes multiples, soit parallèlement soit séquentiellement, à une série d'autres DSA.

3.6.10 opérations d'interrogation à entrées multiples: Opérations de type Directory Search Operations (recherche dans l'Annuaire) comme List (listage) et Search (recherche).

3.6.11 résolution du nom: Processus de localisation d'une entrée par adaptation séquentielle, à un nœud du DIT de chaque nom distinctif relatif (RDN) contenu dans un nom prétendu.

3.6.12 rattachement opérationnel hiérarchique non spécifique (NHOB): Relation entre deux DSA maîtres détenant des contextes de dénomination, dont l'un est immédiatement subordonné à l'autre, où le DSA supérieur détient une référence subordonnée non spécifique au DSA subordonné.

3.6.13 décomposition de référence subordonnée non spécifique (NSSR): Décomposition de références non spécifiques de connaissances en sous-demandes de continuation adressées à des DSA; ces sous-demandes peuvent soit être concaténées avec ces DSA par le DSA exécutant la décomposition ou prendre la forme de références de continuation identifiant les DSA: ces références pourront être retournées au demandeur pour lui permettre de continuer; ou bien le DSA décomposeur peut donner suite à certaines des sous-demandes, laissant les autres sans suite, pour continuation par le demandeur.

3.6.14 avancement d'opération: Série de valeurs qui dénotent la mesure dans laquelle la résolution de nom a eu lieu.

3.6.15 initiateur: DUA qui a lancé une opération spécifique (répartie).

3.6.16 exécuteur: DSA recevant une demande (pour accomplir une opération).

3.6.17 procédure: Spécification (informelle) de la manière dont un DSA applique, à un résultat, une série d'arguments d'entrée et son arbre d'informations de DSA.

NOTE – Les arguments d'entrée et les résultats peuvent correspondre à l'information reçue dans une opération demandée et à l'information envoyée dans une réponse; ils peuvent aussi représenter des étapes intermédiaires dans le calcul d'une réponse suite à une opération demandée. Au 14.2, le premier type d'arguments d'entrée et de résultats est qualifié d'externe.

3.6.18 rattachement opérationnel hiérarchique pertinent (RHOB): Rattachement HOB ou NHOB, selon le contexte.

3.6.19 renvoi de référence: Résultat qui peut être renvoyé par un DSA qui ne peut effectuer l'opération lui-même; ce résultat identifie un ou plusieurs autres DSA ayant davantage la capacité d'effectuer l'opération.

3.6.20 réponse: Résultat ou erreur.

3.6.21 demande: Information consistant en un code d'opération et en arguments associés pour acheminer une opération d'annuaire d'un demandeur à un exécuteur.

3.6.22 décomposition de demande: Décomposition d'une demande en sous-demandes en vue de la poursuite du traitement par d'autres DSA; ces sous-demandes peuvent être concaténées avec les DSA en question par les DSA qui accomplissent la décomposition, ou bien des références de continuation identifiant les DSA peuvent être renvoyées au demandeur en vue de la poursuite du traitement, ou encore le DSA qui procède à la décomposition peut poursuivre le traitement de certaines des sous-demandes, laissant le soin au demandeur de poursuivre le traitement de celles qu'il ignore.

3.6.23 demandeur: DUA ou DSA envoyant une demande d'exécution (c'est-à-dire invoquant) une opération.

3.6.24 opérations d'interrogation à entrée unique: Opérations du type Directory Read Operations, c'est-à-dire Read (lire) et Compare (comparer).

3.6.25 erreur temporaire: Erreur qui peut être passagère ou qui peut déceler un problème localisé, auquel cas l'utilisation d'une référence de connaissances différente ou d'un point d'accès différent peut permettre l'obtention d'un résultat ou d'une erreur permanente.

9594-4 : 1995 (F)

3.6.26 DSA subordonné: De deux DSA partageant un HOB ou un NHOB, celui qui détient le contexte de dénomination subordonné.

3.6.27 sous-demande: Demande obtenue par subdivision d'une demande.

3.6.28 DSA supérieur: De deux DSA partageant un HOB ou NHOB, celui qui détient le contexte de dénomination supérieur.

3.6.29 DSA subordonné: De deux DSA maîtres détenant des contextes de dénomination, celui qui est immédiatement subordonné à l'autre; la relation entre les deux DSA est gérée de manière explicite par l'intermédiaire d'un HOB (ou NHOB), ou existe de manière implicite du fait que le DSA supérieur détient une référence subordonnée (ou subordonnée non spécifique) par rapport au DSA subordonné.

3.6.30 nom d'objet cible: Nom d'une entrée vers laquelle l'opération doit être dirigée à un stade particulier de la résolution du nom ou qui est impliquée dans l'évaluation de cette opération.

3.6.31 unichânage: Mode d'interaction utilisé facultativement par un DSA qui ne peut effectuer lui-même l'opération demandée. Le DSA *chaîne* par invocation d'une opération d'un autre DSA puis par renvoi du résultat au demandeur initial.

4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale les abréviations suivantes sont utilisées:

ASN.1	Notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
DISP	Protocole de duplication miroir des informations d'Annuaire (<i>directory information shadowing protocol</i>)
DMD	Domaine de gestion d'Annuaire (<i>directory management domain</i>)
DOP	Protocole de gestion des rattachements opérationnels pour l'Annuaire (<i>directory operational binding management protocol</i>)
DSE	Entrée spécifique de DSA (<i>DSA specific entry</i>)
HOB	Rattachement opérationnel hiérarchique (<i>hierarchical operational binding</i>)
NHOB	Rattachement opérationnel hiérarchique non spécifique (<i>non-specific hierarchical operational binding</i>)
NSSR	Référence subordonnée non spécifique (<i>non-specific subordinate reference</i>)
RHOB	Rattachement opérationnel hiérarchique pertinent (<i>relevant hierarchical operational binding</i>)

5 Conventions

Sauf exceptions mineures, la présente Spécification d'Annuaire a été élaborée conformément aux directives contenues dans le «Guide pour la coopération entre l'UIT-T et l'ISO/CEI JTC1» et intitulées «Règles de rédaction de texte commun», de mars 1993.

Le terme «Spécification d'Annuaire» (comme dans «la présente Spécification d'Annuaire») a le sens qui lui est attribué dans la Rec. UIT-T X.518 | ISO/CEI 9594-4. Par «Spécifications d'Annuaire» on entendra les Recommandations de la série X.500 et toutes les parties de ISO/CEI 9594.

La présente Spécification d'Annuaire utilise l'expression «systèmes de l'édition 1988», qui fait référence aux systèmes conformes à l'édition précédente (1988) des systèmes de Spécification d'Annuaire c'est-à-dire l'édition 1988 des Recommandations de la série X.500 et l'édition 1990 de ISO/CEI 9594. Ceux qui sont conformes aux actuelles Spécifications d'Annuaire sont appelés «systèmes de l'édition 1993».

Si les éléments d'une liste sont numérotés (et non précédés d'un tiret ou d'une lettre), on considérera que ces éléments sont des étapes d'une marche à suivre.

Les Spécifications d'Annuaire qui définissent des opérations d'Annuaire le font au moyen de la notation des opérations distantes définie dans la Rec. UIT-T X.880 | ISO/CEI 9072-1.

SECTION 2 – VUE D'ENSEMBLE

6 Vue d'ensemble

Le service abstrait d'Annuaire permet l'interrogation, la recherche et la modification des informations de l'Annuaire contenues dans la DIB. Ce service est décrit en termes d'objet abstrait Annuaire, tel que spécifié dans la Rec. UIT-T X.511 | ISO/CEI 9594-3.

De toute évidence, la spécification de l'objet abstrait Annuaire ne saurait concerner la réalisation physique de l'Annuaire; en particulier, elle ne concerne pas la spécification des agents de système d'Annuaire (DSA) dans lesquels sont enregistrées et gérées les informations de la DIB et par l'intermédiaire desquels le service est assuré. En outre, cette spécification ne tient pas compte de la centralisation éventuelle de la DIB, c'est-à-dire du fait qu'elle est contenue dans un seul DSA, ou de sa répartition entre plusieurs DSA. En conséquence, la description du service ne concerne pas non plus les spécifications des DSA en ce qui concerne leurs connaissances des autres DSA, la navigation vers ces agents et la coopération avec eux en vue de la prise en charge du service abstrait dans un environnement réparti.

La présente Spécification d'Annuaire spécifie le raffinement de l'objet abstrait Annuaire, ce raffinement étant exprimé en termes d'ensemble d'un ou de plusieurs objets DSA qui constituent collectivement le service réparti d'Annuaire.

En outre, la présente Spécification d'Annuaire spécifie les modes de répartition autorisés de la base de données DIB entre un ou plusieurs DSA. Dans le cas limite où la base de données DIB est contenue dans un seul DSA, l'Annuaire est en fait centralisé; dans le cas où la DIB est répartie entre deux DSA ou plus, des mécanismes de connaissance et de navigation sont spécifiés: ils garantissent que l'ensemble de la DIB est accessible à tous les DSA qui comportent des entrées constitutives de l'Annuaire.

Des portions de la DIB peuvent aussi être copiées dans plusieurs DSA. Les protocoles décrits dans la présente Spécification d'Annuaire permettent l'emploi de copies d'informations pour l'amélioration de la disponibilité, de la qualité de fonctionnement et de l'efficacité du service d'annuaire réparti. L'emploi d'informations copiées relève dans une certaine mesure de la décision de l'utilisateur, qui peut faire appel à des options de commande de service. Les procédures décrites dans la présente Spécification d'Annuaire indiquent également certaines possibilités d'optimisation de conception avec l'emploi d'informations copiées.

De plus, des interactions de traitement des demandes sont spécifiées: elles permettent aux utilisateurs de l'Annuaire de garder le contrôle sur des caractéristiques particulières de son fonctionnement. En particulier, l'utilisateur peut décider si un DSA a la possibilité, en réponse à une demande présentée à l'Annuaire concernant des informations contenues dans d'autres DSA, d'interroger directement d'autres DSA (chaînage) ou s'il doit fournir en réponse des informations sur d'autres DSA qui pourraient faire progresser le traitement de la demande (renvoi de référence).

En général, la décision d'un DSA d'opérer un chaînage ou un renvoi de référence est déterminée par les commandes de service prévues par l'utilisateur et par les conditions administratives, opérationnelles ou techniques propres au DSA.

Compte tenu du fait qu'en général l'Annuaire sera réparti et que les demandes qui lui seront adressées seront satisfaites par un nombre arbitraire de DSA coopérants qui pourront opérer arbitrairement des chaînages ou des renvois de référence selon les critères ci-dessus, la présente Spécification d'Annuaire spécifie les procédures appropriées qui doivent être suivies par les DSA en réponse à des demandes adressées à l'Annuaire en mode réparti. Ces procédures garantissent que les utilisateurs du service d'Annuaire réparti en auront une perception aussi intuitive que cohérente.

SECTION 3 – MODÈLES D'ANNUAIRE RÉPARTI

7 Modèle du système d'Annuaire réparti

Le service abstrait d'Annuaire, défini dans la Rec. UIT-T X.511 | ISO/CEI 9594-3, modélise l'Annuaire comme un objet qui fournit un ensemble de services d'Annuaire à ses utilisateurs. Les utilisateurs de l'Annuaire accèdent à ses services par un point d'accès. L'Annuaire peut avoir un ou plusieurs points d'accès dont chacun est caractérisé par les services qu'il fournit et par le mode d'interaction utilisé pour les fournir.

La Figure 1 représente le modèle d'Annuaire réparti qui sera utilisé comme base pour la spécification des aspects répartis de l'Annuaire. Elle représente l'objet Annuaire comme étant constitué d'un ensemble d'un ou de plusieurs DSA.

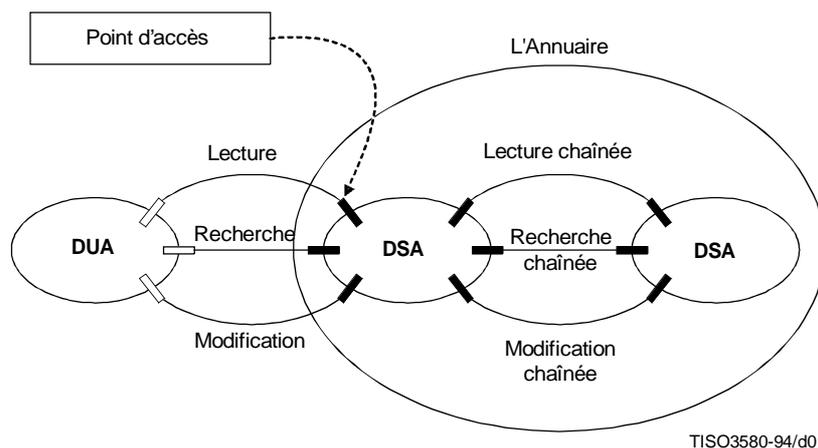


Figure 1 – Objets du modèle d'Annuaire réparti

Les DSA sont spécifiés en détail dans les articles suivants de la présente Spécification d'Annuaire. Cet article énumère simplement un certain nombre de leurs caractéristiques, à titre d'introduction pour préciser les relations entre la présente Spécification d'Annuaire et les autres spécifications de ce type.

Les DSA sont définis en vue de permettre la répartition de la base de données DIB et l'interaction entre plusieurs DSA répartis physiquement, de façon prescrite et coopérative, pour fournir des services d'Annuaire aux utilisateurs de celui-ci (DUA).

La Figure 1 illustre la relation entre le service abstrait d'Annuaire et le service abstrait de DSA. Le service abstrait d'Annuaire défini dans la Rec. UIT-T X.511 | ISO/CEI 9594-3 est fourni au moyen d'un certain nombre d'opérations d'Annuaire. Pour assurer ce service, les DSA qui ont l'Annuaire agissent en interaction. La nature de cette interaction est définie du point de vue du service qu'un DSA peut assurer à un autre DSA, à savoir le service abstrait de DSA. Le service abstrait de DSA est fourni au moyen d'un certain nombre d'opérations appelées opérations chaînées, chacune ayant son pendant dans le service abstrait d'Annuaire. Ainsi, une opération donnée dans le service abstrait d'Annuaire (par exemple lecture), peut nécessiter de la part du DSA assurant le service une interaction avec un ou plusieurs autres DSA à l'aide d'opérations chaînées (par exemple lecture chaînée).

8 Modèle des interactions entre les DSA

Une caractéristique de base de l'Annuaire est qu'un utilisateur, pour une certaine DIB répartie, doit pouvoir faire aboutir une quelconque demande de service (compte tenu des politiques de sécurité, de contrôle d'accès et d'administration), quel que soit le point d'accès auquel cette demande est présentée. Pour répondre à cet impératif, il est nécessaire que tous les DSA qui entrent en compte dans la réponse à une demande de service particulière aient une certaine connaissance (spécifiée dans la Rec. UIT-T X.501 | ISO/CEI 9594-2) de l'endroit où l'information demandée est située, qu'ils retournent cette connaissance au demandeur ou qu'ils essayent, pour le compte de celui-ci, de faire aboutir sa demande. (Le demandeur peut être un DUA ou un autre DSA: dans le dernier cas, les deux DSA doivent supporter le protocole du système d'Annuaire (DSP) (*directory system protocol*)).

Pour répondre à ces impératifs, trois modes d'interaction des DSA sont définis, à savoir l'«unichaînage», le «multichaînage» et le «renvoi de référence». Dans la suite du texte de la présente Spécification d'Annuaire, le terme générique chaînage désignera l'unichaînage et/ou le multichaînage, selon le contexte. Le «chaînage» désigne la tentative faite par un DSA de répondre à une demande en envoyant une ou plusieurs opérations chaînées à d'autres DSA; le «renvoi de référence» désigne le renvoi d'informations de connaissances au demandeur, qui peut ensuite lui-même agir en interaction avec le DSA (ou les DSA) identifiés dans les informations de connaissances.

Une interaction par unichaînage ou par renvoi de référence peut faire suite à une demande isolée. En variante, la demande peut être décomposée en plusieurs sous-demandes avant l'interaction. Des interactions par multichaînage ou par renvois de référence, ou une combinaison des deux modes, peuvent faire suite à une demande décomposée. Deux types de décomposition sont définis: la décomposition de NSSR et la décomposition de demande.

8.1 Décomposition d'une demande

8.1.1 Décomposition NSSR

La décomposition NSSR consiste à préparer des demandes identiques, prêtes au transfert (soit en séquence ou en parallèle) vers plusieurs DSA subordonnés, à la suite de la rencontre d'une NSSR au cours de la résolution du nom. Comme les références subordonnées non spécifiques ne contiennent pas les RDN des contextes de dénomination subordonnés appelés en référence, le DSA attribuant les références n'est pas en mesure de déterminer le DSA subordonné qui contient le ou les contexte(s) de dénomination subordonné(s). Au cours de la résolution du nom, un DSA qui rencontre des NSSR doit envoyer une demande identique à chaque DSA subordonné (en l'absence de duplication miroir). Cela peut être fait en séquence ou en parallèle. Généralement, un seul DSA sera en mesure de poursuivre la résolution du nom et les autres renverront l'erreur de service **unableToProceed** (traitement impossible). Dans certaines (rares) circonstances, il se peut que plus d'un DSA poursuive la résolution du nom, ce qui donne lieu à des résultats dupliqués.

8.1.2 Décomposition de demande

La décomposition de demande, autre façon de décomposer une demande, est un processus interne effectué par un DSA avant la communication avec un ou plusieurs autres DSA. Une demande est décomposée en plusieurs sous-demandes, si possible différentes, de telle manière que chacune de ces sous-demandes accomplisse une partie de la tâche initiale. La décomposition de demande ne peut être utilisée qu'au cours de l'évaluation d'une opération de listage (List) ou de recherche (Search). Après décomposition de la demande, chacune des sous-demandes peut être chaînée à d'autres DSA afin de poursuivre la tâche, ou un résultat partiel (renvoi de référence encadré) peut être renvoyé au demandeur. Un exemple de production d'une même sous-demande à destination de DSA (différents) se présente quand une entrée a des références subordonnées et/ou des NSSR qui renvoient ensemble à plusieurs DSA. Un exemple de sous-demandes différentes renvoyées aux mêmes DSA ou à des DSA différents se présente quand deux entrées différentes sont rencontrées au cours d'une recherche (sous-arbre), et que chacune a une référence subordonnée.

8.2 Unichaînage

Ce mode d'interaction (illustré à la Figure 2) peut être utilisé par un DSA pour communiquer une demande à un autre DSA, quand le premier a connaissance des contextes de dénomination détenus par le second. L'unichaînage peut être utilisé pour prendre contact avec un seul DSA désigné dans une référence croisée, une référence subordonnée, une référence supérieure, une référence de fournisseur ou une référence principale.

NOTE – Sur la Figure 2, l'ordre des interactions est défini par les numéros associés aux flèches.

8.3 Multichaînage

Ce mode d'interaction est utilisé par un DSA pour transférer plusieurs demandes sortantes qui résultent d'une demande entrante, par suite de décomposition de demande ou de NSSR.

8.3.1 Multichaînage parallèle

En multichaînage parallèle, le DSA transfère simultanément plusieurs demandes sortantes (voir la Figure 3a). Si le multichaînage parallèle peut améliorer la qualité de fonctionnement, il peut aussi, dans certaines circonstances telles que la duplication miroir, entraîner la réception de résultats en double.

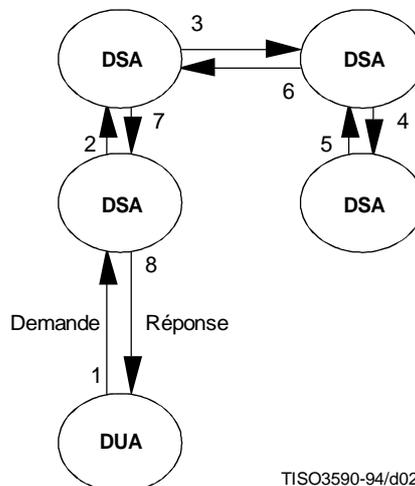


Figure 2 – Mode unichaînage

9594-4 : 1995 (F)

8.3.2 Multichaînage séquentiel

En multichaînage séquentiel, le DSA transfère une seule demande sortante à la fois et attend le résultat (ou l'erreur) correspondant à cette demande avant d'envoyer la suivante (voir la Figure 3b). Si le multichaînage séquentiel n'est pas le mode d'interaction le plus rapide, il est cependant peu probable qu'il entraîne la réception de résultats en double.

NOTE – Un DSA peut utiliser une combinaison de multichaînage parallèle et de multichaînage séquentiel.

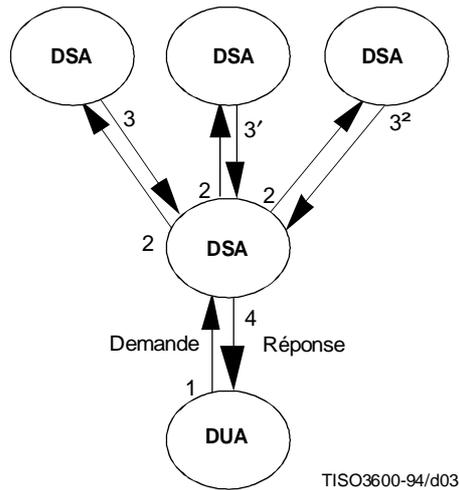
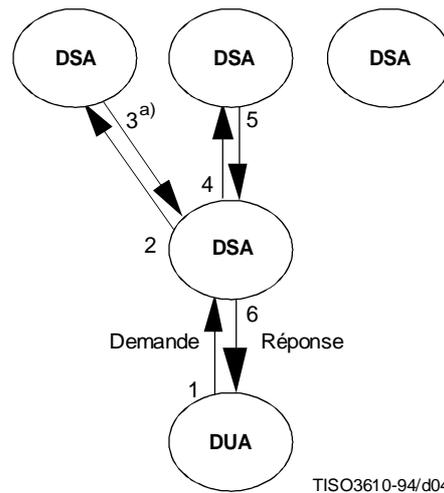


Figure 3a – Multichaînage parallèle



a) Traitement impossible.

Figure 3b – Multichaînage séquentiel (à la suite d'une décomposition de NSSR)

8.4 Renvoi de référence

Un renvoi de référence (illustré aux Figures 4a et 4b) est retourné par un DSA dans la réponse à une demande issue soit d'un DUA ou d'un autre DSA. Le renvoi de référence peut constituer toute la réponse (auquel cas il est considéré comme une erreur) ou seulement une partie de la réponse. Le renvoi de référence contient une référence de connaissance, qui peut être une référence supérieure, subordonnée, croisée, subordonnée non spécifique, de fournisseur ou principale.

Le DSA (Figure 4a) recevant le renvoi de référence peut utiliser la référence de connaissance qui y est contenue pour ensuite chaîner ou multichaîner (selon le type de référence) la demande initiale vers d'autres DSA. Une autre possibilité serait qu'un DSA recevant un renvoi de référence le retourne à son tour dans sa réponse. Un DUA (Figure 4b) recevant un renvoi de référence peut l'utiliser pour prendre contact avec un ou plusieurs autres DSA afin de faire progresser la demande.

NOTE – Sur les Figures 4a et 4b, l'ordre des interactions est défini par les numéros associés aux flèches.

8.5 Détermination du mode

Si un DSA ne peut pas satisfaire entièrement lui-même une demande, il doit chaîner cette demande (ou une demande résultant de la décomposition de la demande initiale) vers un autre DSA, sauf si:

- a) le chaînage est interdit par l'utilisateur via les commandes de service, auquel cas le DSA doit retourner un renvoi de référence ou une erreur de type **chainingRequiredServiceError**; ou
- b) le DSA a des raisons d'ordre administratif, opérationnel ou technique de préférer le non-chaînage, auquel cas le DSA doit retourner un renvoi de référence.

NOTES

- 1 Une «raison d'ordre technique» interdisant le chaînage est le fait que le DSA identifié dans la référence de connaissance ne gère pas le protocole DSP.
- 2 Si la commande de service **localScope** est activée, le DSA (ou le domaine de gestion d'annuaire (DMD) doit soit résoudre la demande, soit retourner une erreur.
- 3 Si l'utilisateur préfère les renvois de référence, il active la commande **chainingProhibited**.

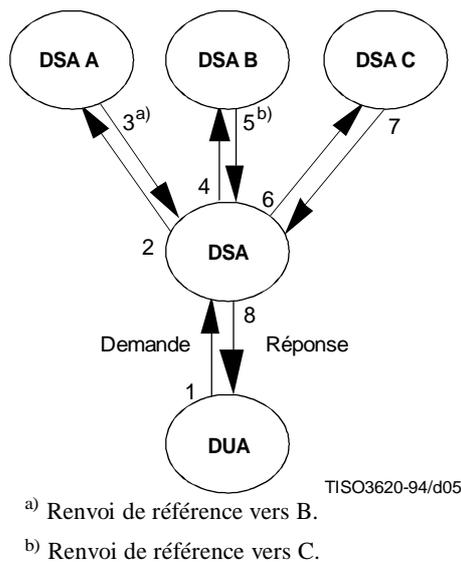
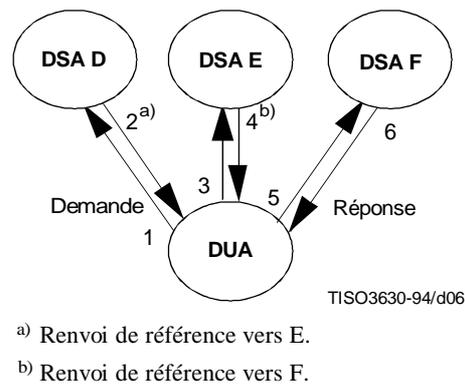


Figure 4a – Mode renvoi de référence (action de DSA sur les renvois de référence)



**Figure 4b – Mode renvoi de référence
(action de DSA sur des renvois de référence)**

SECTION 4 – SERVICE ABSTRAIT DE DSA

9 Vue d'ensemble du service abstrait de DSA

Le service d'Annuaire est entièrement décrit dans la Rec. UIT-T X.511 | ISO/CEI 9594-3. Quand un tel service est fourni dans un environnement réparti, selon le modèle décrit à l'article 7, il peut être considéré comme fourni au moyen d'un ensemble de DSA. Cela est représenté sur la Figure 1.

Pour chaque opération définie dans le service d'Annuaire, une opération «chaînée» correspondante est définie dans le service abstrait de DSA en vue d'être utilisée entre les DSA qui contribuent à l'accomplissement de cette opération du service d'Annuaire. Ainsi, un DSA recevant une opération lecture de la part d'un DUA pourrait nécessiter l'assistance d'un autre DSA (par exemple un DSA détenant l'entrée cible ou une copie de cette entrée) pour en assurer l'accomplissement et envoyer ainsi à ce DSA une opération lecture chaînée.

Les types d'information échangés dans le service abstrait de DSA sont définis à l'article 10. Les opérations et les erreurs du service abstrait de DSA sont définies aux articles 11 à 13.

10 Types d'information

10.1 Introduction

Cet article identifie et définit parfois un certain nombre de types d'information qui seront utilisés par la suite pour la définition de diverses opérations du service abstrait de DSA. Les types d'information concernés sont ceux qui sont communs à plusieurs opérations, qui ont des chances d'être utilisés à l'avenir, ou qui sont suffisamment complexes ou autonomes pour mériter d'être définis séparément des opérations qui les utilisent.

Plusieurs des types d'information utilisés dans la définition du service abstrait de DSA sont en fait définis ailleurs. Le paragraphe 10.2 signale ces types et indique l'origine de leur définition. Chacun des paragraphes suivants (10.3 à 10.10) identifie et définit un type d'information.

10.2 Types d'information définis ailleurs

Les types d'information suivants sont définis dans la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- **aliasedEntryName;**
- **DistinguishedName;**
- **Name;**
- **RelativeDistinguishedName.**

Les types d'information suivants sont définis dans la Rec. UIT-T X.511 | ISO/CEI 9594-3:

(Rattachement)

- **DirectoryBind**

(Opérations)

- **Abandon**

(Erreurs)

- **abandoned;**
- **attributeError;**
- **nameError;**
- **securityError;**
- **serviceError;**
- **updateError.**

(Classe d'objets d'information)

- **OPTIONALLY-SIGNED**

(Type de Données)

- **SecurityParameters**

Le type d'information suivant est défini dans la Rec. UIT-T X.520 | ISO/CEI 9594-6:

- **PresentationAddress.**

10.3 Arguments de chaînage

Le composant **ChainingArguments** figure dans chaque opération chaînée pour véhiculer vers un DSA les informations nécessaires au succès de l'exécution de sa partie de la tâche totale:

```

ChainingArguments ::= SET {
    originator           [0] DistinguishedName OPTIONAL,
    targetObject         [1] DistinguishedName OPTIONAL,
    operationProgress    [2] OperationProgress
                        DEFAULT { nameResolutionPhase notStarted },
    traceInformation     [3] TraceInformation,
    aliasDereferenced    [4] BOOLEAN DEFAULT FALSE,
    aliasedRDNs          [5] INTEGER OPTIONAL,
                        -- absent sauf si aliasDereferenced est TRUE
    returnCrossRefs      [6] BOOLEAN DEFAULT FALSE,
    referenceType        [7] ReferenceType DEFAULT superior,
    info                 [8] DomainInfo OPTIONAL,
    timeLimit            [9] UTCTime OPTIONAL,
    securityParameters   [10] SecurityParameters DEFAULT { },
    entryOnly            [11] BOOLEAN DEFAULT FALSE,
    uniqueIdentifier     [12] UniqueIdentifier OPTIONAL,
    authenticationLevel  [13] AuthenticationLevel OPTIONAL,
    exclusions           [14] Exclusions OPTIONAL,
    excludeShadows      [15] BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster  [16] BOOLEAN DEFAULT FALSE }
    
```

9594-4 : 1995 (F)

Les divers composants ont le sens défini ci-après.

- a) Le composant **originator** véhicule le nom du (dernier) initiateur de la demande, à moins qu'il ne soit déjà spécifié dans les paramètres de sécurité. Si le composant **requester** est présent dans le composant **CommonArguments**, cet argument peut être omis.
- b) Le composant **targetObject** véhicule le nom de l'objet dont l'entrée d'annuaire est la cible. Le rôle de cet objet dépend de l'opération concernée: il peut s'agir de l'objet dont l'entrée est la cible ou de l'entrée d'objet qui doit devenir la base pour une demande ou sous-demande impliquant plusieurs objets (par exemple **ChainedList** ou **ChainedSearch**). Ce composant ne peut être omis que s'il a la même valeur que le paramètre d'objet ou d'entrée d'objet de base dans l'opération chaînée, auquel cas sa valeur implicite sera cette valeur.
- c) Le composant **operationProgress** s'utilise pour informer le DSA du déroulement de l'opération et donc du rôle que ce DSA est supposé jouer dans l'exécution de l'opération complète. Les informations véhiculées dans ce composant sont spécifiées au 10.5.
- d) Le composant **traceInformation** est utilisé pour éviter le bouclage entre des DSA lorsque le chaînage est activé. Un DSA ajoute un nouvel élément à l'information de trace avant de chaîner une opération vers un autre DSA. Quand il reçoit une demande d'exécution d'une opération, un DSA vérifie, en examinant l'information de trace, que cette opération n'a pas formé de boucle. Les informations véhiculées dans ce composant sont spécifiées au 10.6.
- e) Le composant **aliasDereferenced** est une valeur de type **BOOLEAN** utilisée pour indiquer si une ou plusieurs entrées alias ont déjà été rencontrées ou non et si leur nom a été échangé au cours de la résolution répartie du nom. La valeur par défaut **FALSE** indique qu'aucune entrée alias n'a été déréférencée.
- f) Le composant **aliasedRDNs** indique combien de RDN contenus dans le composant **targetObject Name** ont été émis à partir des attributs **aliasedEntryName** d'une (ou de plusieurs) entrées alias. La valeur de l'entier est établie chaque fois qu'une entrée alias est rencontrée et déréférencée. Ce composant ne doit figurer que si et seulement si le composant **aliasDereferenced** a la valeur **TRUE**.
- g) Le composant **entryOnly** est mis à **TRUE** si l'opération initiale était une recherche (l'argument de sous-ensemble étant mis à **oneLevel**) et qu'un alias ait été rencontré en tant que subordonné immédiat de l'entrée d'objet de base (**baseObject**). Le DSA qui réussit à effectuer la résolution du nom de l'objet cible (**targetObject**) n'effectuera l'évaluation de cet objet que sur l'entrée nommée.
- h) Le composant **returnCrossRefs** est une valeur booléenne qui indique si des références de connaissance, utilisées au cours de l'exécution d'une opération répartie, font ou non l'objet d'une demande de retour au DSA initial en tant que références croisées associées à un résultat ou à un renvoi de référence. La valeur par défaut **FALSE** indique que ces références de connaissance n'ont pas à être retournées.
- i) Le composant **referenceType** indique au DSA auquel il est demandé d'exécuter l'opération, quel type d'information de connaissance a été utilisé pour acheminer la demande vers lui. Le DSA peut ainsi détecter des erreurs dans les informations de connaissance détenues par le demandeur. Si une telle erreur est détectée, elle doit être indiquée par un message **ServiceError**, avec la cause **invalidReference**. Le composant **ReferenceType** est décrit en détail au 10.7.

NOTE – Si le composant **referenceType** manque, la valeur «superior» est implicite.

- j) Le composant **Info** est utilisé pour véhiculer des informations spécifiques au domaine de gestion d'Annuaire (DMD) entre les DSA qui sont impliqués dans le traitement en commun d'une demande. Ce composant est du type **DomainInfo**, qui est non restreint:

DomainInfo ::= **ABSTRACT-SYNTAX.&Type**

- k) Le composant **timeLimit**, s'il est présent, indique le délai d'exécution de l'opération [voir 17.4.4b)].
- l) Le composant **SecurityParameters** est spécifié dans la Rec. UIT-T X.511 | ISO/CEI 9594-3. Son absence équivaut à un ensemble vide de paramètres de sécurité.
- m) Le composant **AuthenticationLevel** est fourni facultativement quand il est nécessaire d'indiquer la manière dont l'authentification a été effectuée. Il est décrit dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.
- n) Le composant **UniqueIdentifier** est fourni facultativement quand il est nécessaire de confirmer le nom de l'initiateur. Il est décrit dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.

- o) Le composant **entryOnly** est mis à **TRUE** si l'opération initiale était une recherche (l'argument **subset** étant mis à **oneLevel**) et qu'une entrée alias ait été rencontrée en tant que subordonnée immédiate de l'entrée d'objet de base (**baseObject**). Le DSA qui réussit à effectuer la résolution du nom de l'objet cible (**targetObject**) n'effectuera l'évaluation de cet objet que sur l'entrée nommée.
- p) Le composant **exclusions** n'a de signification que pour les opérations de recherche; s'il est présent, il indique les sous-arbres d'entrées subordonnées à l'objet cible (**targetObject**) qui seront exclues du résultat d'une opération de recherche.
- q) Le composant **excludeShadows** n'a de sens que pour les opérations de recherche et de listage. Il indique que la recherche doit être appliquée aux entrées et non pas aux copies d'entrées. Ce composant facultatif peut être utilisé par un DSA pour éviter la duplication des résultats reçus (voir 20.1).
- r) Le composant **nameResolveOnMaster** n'a de sens que pendant la résolution du nom et est uniquement sélectionné si des NSSR ont été rencontrées. Avec la valeur **TRUE**, il indique que la résolution du nom ultérieure, c'est-à-dire l'adaptation des RDN qui restent à partir de **nextRDNTToBeResolved**, ne doit pas utiliser l'information de copie d'entrée; la résolution ultérieure de chaque RDN restant doit être effectuée dans le DSA maître pour l'entrée identifiée par ce RDN (voir 20.1).

10.4 Résultats du chaînage

Le composant **ChainingResults** figure dans le résultat de chaque opération et fournit des informations en retour au DSA qui a invoqué cette opération.

```
ChainingResults ::= SET {
  info [0] DomainInfo OPTIONAL,
  crossReferences [1] SEQUENCE OF CrossReference OPTIONAL,
  securityParameters [2] SecurityParameters DEFAULT {},
  alreadySearched [3] Exclusions OPTIONAL }
```

Les divers composants ont les significations définies ci-après:

- a) le composant **Info** est utilisé pour véhiculer des informations spécifiques à un DMD entre des DSA qui sont impliqués dans le traitement d'une demande commune. Ce composant est du type **DomainInfo**, qui est non restreint;
- b) le composant **crossReferences** ne figure pas dans le composant **ChainingResults**, sauf si le composant **returnCrossRefs** de la demande correspondante à la valeur **TRUE**. Ce composant comprend une séquence d'items **CrossReference** dont chacun contient un **contextPrefix** et un descripteur de point d'accès (**accessPoint**) (voir 10.8).

```
CrossReference ::= SET {
  contextPrefix [0] DistinguishedName,
  accessPoint [1] AccessPointInformation }
```

Un composant **CrossReference** peut être ajouté par un DSA quand il trouve une correspondance entre une partie de l'argument **targetObject** d'une opération et un de ses préfixes de contexte. L'autorité administrative d'un DSA peut avoir pour politique de ne pas retourner une telle information de connaissance: dans ce cas, elle n'ajoutera pas d'items à la séquence.

- c) Le composant **SecurityParameters** est spécifié dans la Rec. UIT-T X.511 | ISO/CEI 9594-3. Lorsque ce paramètre est absent, on considère qu'il existe un ensemble vide de paramètres de sécurité.
- d) Le composant **alreadySearched**, s'il est présent indique quels RDN immédiatement subordonnés au composant **targetObject** ont été traités dans le cadre d'une opération de recherche chaînée. Il doit donc être exclu lors d'une sous-demande subséquente.

10.5 Déroulement de l'opération

Une valeur **OperationProgress** décrit l'état d'avancement de l'exécution d'une opération à laquelle doivent participer plusieurs DSA.

```
OperationProgress ::= SET {
  nameResolutionPhase [0] ENUMERATED {
    notStarted (1),
    proceeding (2),
    completed (3) },
  nextRDNTToBeResolved [1] INTEGER OPTIONAL }
```

9594-4 : 1995 (F)

Les divers composants ont le sens défini ci-après.

- a) Le composant **nameResolutionPhase** indique la phase atteinte dans le traitement du nom d'une opération de type **targetObject**. Quand il indique que la résolution du nom a la valeur **notStarted**, cela signale qu'il n'a pas encore été possible d'atteindre un DSA détenant un contexte de dénomination contenant le ou les RDN initiaux du nom. Si la résolution du nom a la valeur **proceeding**, cela indique que la partie initiale du nom a été reconnue, mais que le DSA contenant l'objet cible n'a pas encore été atteint. Le composant **NextRDNTToBeResolved** indique quelle fraction du nom a déjà été reconnue [voir 10.5 b)]. Si la résolution du nom a la valeur **completed**, le DSA contenant le composant **targetObject** a été atteint et l'exécution de l'opération proprement dite est en cours.
- b) Le composant **nextRDNTToBeResolved** indique au DSA quel est le prochain des RDN contenus dans le composant **targetObject** qu'il faudra résoudre. Il a la forme d'un entier compris entre 1 et le nombre de RDN du nom. Ce composant n'est présent que si le composant **nameResolutionPhase** a la valeur **proceeding**.

10.6 Information de trace

Une valeur **TraceInformation** véhicule un enregistrement des DSA qui ont déjà participé à l'exécution de l'opération. Elle est utilisée pour détecter l'existence de bouclages, ou pour les éviter car ces bouclages peuvent provenir de l'incohérence d'informations de connaissance ou de la présence de bouclages d'alias dans le DIT.

TraceInformation	::=	SEQUENCE OF TraceItem
TraceItem	::=	SET {
dsa	[0]	Name,
targetObject	[1]	Name OPTIONAL,
operationProgress	[2]	OperationProgress }

Chaque DSA qui renvoie une opération vers un autre agent ajoute un nouvel élément à la fin de la séquence des composants **TraceItem**. Chacun de ces composants **TraceItem** contient:

- a) le nom de **dsa** qui ajoute l'item;
- b) le nom du composant **targetObject** que le DSA ajoutant l'item a reçu avec la demande entrante. Ce paramètre est omis si sa demande à chaîner provient d'un DUA (auquel cas sa valeur implicite est le composant **object** ou **baseObject** dans **XOperation**) ou si sa valeur est la même que celle du composant **targetObject** (réelle ou implicite) contenu dans l'argument **ChainingArgument** de la demande sortante;
- c) la valeur **operationProgress** que le DSA ajoutant l'item a reçue au sujet de la demande entrante.

10.7 Type de référence

Une valeur **ReferenceType** indique un des divers types de référence définis dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.

ReferenceType	::=	ENUMERATED {
superior	(1),	
subordinate	(2),	
cross	(3),	
nonSpecificSubordinate	(4),	
supplier	(5),	
master	(6),	
immediateSuperior	(7),	
self	(8) }	

10.8 Information sur le point d'accès

Il y a trois types de points d'accès.

- a) Une valeur de point d'accès (**AccessPoint**) identifie un point particulier où peut s'effectuer un accès à l'Annuaire, plus précisément à un DSA. Le point d'accès a un nom (**Name**), celui du DSA concerné et une adresse de présentation (**PresentationAddress**), à utiliser lors des communications OSI vers ce DSA.

AccessPoint	::=	SET {
ae-title	[0]	Name,
address	[1]	PresentationAddress,
protocolInformation	[2]	SET OF ProtocolInformation OPTIONAL }

- b) Une valeur **MasterOrShadowAccessPoint** identifie un point d'accès à l'Annuaire. Le composant **category**, (de valeur **master** ou **shadow**) du point d'accès est différent selon que ce point est orienté vers un contexte de dénomination ou vers une zone copiée d'utilisation commune.

```

MasterOrShadowAccessPoint ::= SET {
  COMPONENTS OF
  category [3] ENUMERATED {
    master (0),
    shadow (1) } DEFAULT master }
    
```

- c) Une valeur **MasterAndShadowAccessPoints** identifie une série de points d'accès à l'Annuaire, à savoir un groupe de DSA en relation. Ces points d'accès ont la caractéristique commune que chacun se réfère à un DSA détenant une information d'entrée provenant d'un contexte de dénomination commun (ou d'un tronçon commun de contextes de dénomination détenus par un même DSA quand la valeur est celle de l'attribut **nonSpecificKnowledge**). Une valeur **MasterAndShadowAccessPoints** indique le composant **category** de chaque valeur **AccessPoint** qu'elle contient. Le point d'accès du DSA maître du contexte de dénomination ne doit pas nécessairement faire partie de l'ensemble.

```

MasterAndShadowAccessPoints ::= SET OF MasterOrShadowAccessPoint
    
```

Une valeur **AccessPointInformation** identifie un ou plusieurs points d'accès à l'Annuaire.

```

AccessPointInformation ::= SET {
  COMPONENTS OF
  additionalPoints [4] SET OF MasterOrShadowAccessPoint OPTIONAL }
    
```

Si des DSA édition 1988 produisent une valeur **AccessPointInformation**, la composante facultative de l'ensemble est absente. Lorsque des DSA édition 1988 interprètent une valeur **AccessPointInformation**, toute valeur **MasterAndShadowAccessPoint** présente est ignorée.

Dans le cas des DSA édition 1993, la composante de valeur **MasterOrShadowAccessPoint**, produite pour une valeur **AccessPointInformation**, peut appartenir à la catégorie maître ou à la catégorie duplication miroir, selon ce qui est déterminé par la procédure de sélection de connaissance du DSA produisant la valeur. On peut la considérer comme un point d'accès suggéré fourni par le DSA qui produit la valeur à l'intention du DSA qui la reçoit. Une série de valeurs **MasterAndShadowAccessPoints** peut aussi être (facultativement) produite pour une valeur **AccessPointInformation**. Cela constitue une information supplémentaire qui pourra être utilisée par la procédure de sélection de connaissance du DSA récepteur pour déterminer un point d'accès de rechange.

10.9 Exclusions

Comme cela est défini au 10.3, le composant **exclusions** de **ChainingArguments** s'utilise pour limiter la portée de l'opération de recherche en identifiant un certain nombre d'entrées subordonnées à l'entrée de l'objet cible qui, avec toutes leurs autres subordonnées, ne doivent pas faire partie du traitement de l'opération de recherche. Le composant **exclusion** est défini comme une valeur du type **Exclusions** de l'ASN.1.

```

Exclusions ::= SET OF RDNSequence
    
```

Chaque valeur **RDNSequence** du groupe **Exclusions** doit normalement identifier le préfixe du contexte d'un subordonné du contexte de dénomination à l'entrée de l'objet cible. Si un DSA reçoit une demande de recherche avec une valeur **RDNSequence** n'obéissant pas à cette contrainte, ce DSA peut ignorer la valeur en question. La valeur **RDNSequence** se rapporte à l'objet cible et n'est pas le nom distinctif du préfixe de contexte.

Les **Exclusions** peuvent, non seulement faire partie d'une demande d'utilisateur mais aussi être utilisées par des DSA pour minimiser les informations en double provenance de sous-demandes de recherche émises en présence d'informations miroirs.

La Figure 5 illustre un exemple de l'utilisation du type **Exclusions**. Dans cet exemple, un DSA détient deux zones copiées, l'une en dessous de l'autre. L'une débute par le préfixe de contexte X, l'autre par le préfixe de contexte C. Une copie d'entrée à Y possède trois références subordonnées à trois contextes de dénomination A, B et C.

Si, à titre d'exemple, on effectue pour ce DSA une recherche dans un sous-arbre débutant par une entrée d'objet de base contenue dans un contexte de dénomination X, le DSA peut fournir des informations en provenance des zones copiées X et C. Les informations en provenance des contextes de dénomination A et B peuvent être fournies via les références subordonnées. A la décomposition de la demande, le composant **ContinuationReferences**, qu'il faudra utiliser dans des résultats partiels (**partialResults**) ou dans un chaînage, spécifiera Y comme l'objet cible et C comme l'élément unique d'un ensemble de type **Exclusions**.

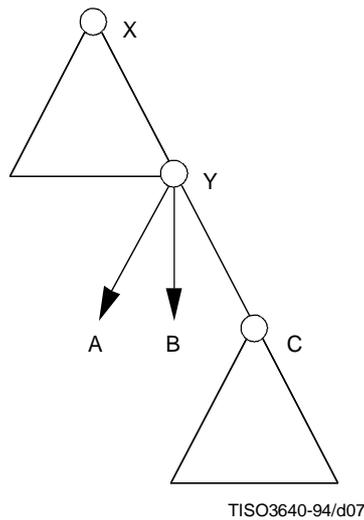


Figure 5 – Exclusions

10.10 Référence de continuation

Le composant **ContinuationReference** décrit comment l'exécution de l'ensemble ou d'une partie d'une opération peut être poursuivie par un ou plusieurs DSA différents. Elle est normalement retournée comme renvoi de référence quand le DSA concerné est incapable de faire suivre la demande lui-même, ou ne le désire pas.

```
ContinuationReference ::= SET {
    targetObject      [0] Name,
    aliasedRDNs      [1] INTEGER OPTIONAL,
    operationProgress [2] OperationProgress,
    rdnsResolved     [3] INTEGER OPTIONAL,
    referenceType     [4] ReferenceType ,
    accessPoints     [5] SET OF AccessPointInformation,
    entryOnly        [6] BOOLEAN DEFAULT FALSE,
    exclusions       [7] Exclusions OPTIONAL,
    returnToDUA     [8] BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster [9] BOOLEAN DEFAULT FALSE}
```

Ces divers composants ont le sens défini ci-après:

- a) Le composant **targetObject Name** indique le nom qui est proposé pour la suite de l'opération. Il peut être différent du nom **targetObject Name** reçu dans la demande entrante si, par exemple un alias a été déréférencé ou si l'entrée d'objet de base d'une recherche a été localisée.
- b) Le composant **aliasedRDNs** indique combien de RDN éventuellement contenus dans le nom de l'objet cible, résultent de déréférencements d'alias. L'argument n'apparaît que si un nom d'alias a été déréférencé.
- c) Le composant **operationProgress** indique l'état d'avancement de la résolution du nom qui a été réalisé et qui déterminera la suite de l'exécution de l'opération par les DSA nommés, à condition que le DSA ou le DUA recevant la référence **ContinuationReference** souhaite lui donner suite.
- d) La valeur du composant **rdnsResolved** (qui doit apparaître que si certains des RDN du nom n'ont pas été pleinement résolus, mais ont été considérés comme corrects en vertu d'une référence croisée) indique combien de RDN ont été effectivement résolus, en utilisant uniquement des références internes.
- e) Le composant **referenceType** indique le type d'informations de connaissance qui a été utilisé pour produire cette référence de continuation.
- f) Le composant **accessPoints** indique les points d'accès qu'il faut contacter pour réaliser cette continuation. Il ne peut y avoir plus d'un item **AccessPointInformation** qu'en présence de références subordonnées non spécifiques.

- g) Le composant **entryOnly** est mis à la valeur **TRUE** si l'opération originale était une recherche avec l'argument **subset** mis à **oneLevel** et qu'une entrée alias ait été rencontrée en tant que subordonné immédiat de l'objet **baseObject**. Le DSA qui exécute sans problème la résolution de nom au sujet du nom **targetObject** doit effectuer l'évaluation d'objet sur la seule entrée nommée.
- h) Le composant **exclusions** identifie une série de contextes de dénomination subordonnés qui ne doivent pas être explorés par le DSA récepteur.
- i) L'élément **returnToDUA** est fourni facultativement quand le DSA utilisant la référence de continuation souhaite indiquer qu'il ne veut pas retourner des informations via un DSA intermédiaire (pour des raisons de sécurité par exemple) et qu'il désire faire savoir que ces informations peuvent être directement obtenues par une opération du protocole d'accès à l'Annuaire (DAP) entre le DUA initiateur et le DSA. Lorsque l'élément **returnToDUA** est mis à **TRUE**, le composant **referenceType** peut être mis à **self**.
- j) Sur option, l'élément **nameResolveOnMaster** est fourni lorsque le DSA qui crée la référence de continuation a rencontré des références de type NSSR. Si cet élément est mis à **TRUE**, il signale que la résolution de nom suivante (correspondant aux RDN restants, indiqués par le composant **nextRDNTToBeResolved**) ne doit pas employer d'informations de copie d'entrée; la résolution de nom subséquente doit être faite – pour chaque RDN restant à résoudre – au niveau du DSA maître pour l'entrée identifiée par ce RDN (voir 20.1).

11 Rattachement et détachement

Les opérations **DSABind** et **DSAUnbind** sont respectivement utilisées par un DSA, au début et à la fin d'une période d'accès à un autre DSA.

11.1 DSA Bind (rattachement de DSA)

L'opération **DSABind** s'utilise pour commencer une période de coopération entre deux DSA assurant le service d'Annuaire.

DSABind ::= BIND
ARGUMENT **DirectoryBindArgument**
RESULT **DirectoryBindResult**
BIND-ERROR **DirectoryBindError**

Les composants de la **DSABind** sont identiques à leurs homologues dans **DirectoryBind** (voir la Rec. UIT-T X.511 | ISO/CEI 9594-3) sauf sur les points suivants:

- le composant **Credentials** (accréditation) de l'argument **DirectoryBindArgument** permet d'envoyer au DSA répondeur des renseignements identifiant le titre AE du DSA initiateur. Ce titre d'entité d'application doit avoir la forme d'un nom distinctif d'Annuaire;
- le composant **Credentials** du résultat **DirectoryBindResult** permet d'envoyer au DSA initiateur des renseignements identifiant le titre AE du DSA répondeur. Ce titre d'entité d'application doit avoir la forme d'un nom distinctif.

11.2 DSA Unbind (Détachement de DSA)

L'opération **DSAUnbind** s'utilise pour mettre fin à la coopération entre deux DSA assurant le service d'Annuaire.

DSAUnbind ::= UNBIND

Il n'y a ni arguments, ni résultats ni erreurs.

12 Opérations chaînées

Pour chacune des opérations utilisées pour accéder au service d'Annuaire, il existe une opération utilisée entre des DSA qui coopèrent au moyen d'une correspondance bilatérale. Les noms des opérations ont été choisis de manière à refléter cette correspondance, c'est-à-dire qu'on a donné le préfixe «chained» aux noms des opérations utilisées entre DSA coopérants.

9594-4 : 1995 (F)

Les arguments, résultats et erreurs des opérations chaînées sont, à une exception près, formés systématiquement à partir des arguments, résultats et erreurs des opérations correspondantes du service d'Annuaire (voir 14.3). Cette exception est l'opération **ChainedAbandon**, qui est syntaxiquement équivalente à son homologue du service d'Annuaire (voir 14.4).

12.1 Opérations chaînées

Un DSA qui a reçu d'un DUA une opération peut choisir de construire une forme chaînée de cette opération afin de la communiquer à un autre DSA. Un DSA qui a reçu une forme chaînée d'une opération peut également choisir de la concaténer pour l'envoyer à un autre DSA. Le DSA qui invoque une forme chaînée d'une opération peut, s'il le désire, signer l'argument de cette opération; le DSA exécutant l'opération peut, sur demande, signer le résultat de cette opération.

La forme chaînée d'une opération est spécifiée au moyen du type paramétrisé **chained {}**.

```
chained { OPERATION : operation } OPERATION ::= {
  ARGUMENT OPTIONALLY-SIGNED { SET {
    chainedArgument          ChainingArguments,
    argument                  [0] operation.&ArgumentType } }
  RESULT OPTIONALLY-SIGNED { SET {
    chainedResult           ChainingResults,
    result                   [0] operation.&ResultType } }
  ERRORS { operation.&Errors EXCEPT (referral | dsaReferral) }
  CODE    operation.&operationCode }
```

NOTES

1 Les opérations du service abstrait d'Annuaire que l'on peut utiliser comme paramètre associé au type **chained {}** comprennent l'erreur **abandoned**. La présence de cette erreur faisant partie de l'ensemble des erreurs possibles d'une opération chaînée reflète la possibilité examinée au 14.4 qu'un item **ChainedAbandon** puisse être produit pour une opération **ChainedModify** quand une association liée n'aboutit pas.

2 La spécification définitive du service abstrait de DSA (voir l'Annexe A) applique ce type paramétrisé pour construire toutes les opérations chaînées du service abstrait.

L'argument de l'opération dérivée contient les composants suivants:

- a) Le composant **chainedArgument** – C'est une valeur de l'élément **chainingArguments** contenant les informations (en plus et au-dessus de l'argument initial fourni par le DUA) qui sont nécessaires au DSA exécutant pour effectuer l'opération. Ce type d'information est défini au 10.3.
- b) Le composant **argument** – C'est une valeur de l'élément **operation.&Argument** qui contient l'argument initialement fourni par le DUA, tel que spécifié à l'article approprié de la Rec. UIT-T X.511 | ISO/CEI 9594-3.

Si la demande aboutit, le résultat sera retourné. Les paramètres du résultat auront le sens suivant:

- a) Le composant **chainedResults** – C'est une valeur de l'élément **ChainingResults** qui contient les informations, en plus et au-dessus de celles qu'il faut fournir au DUA demandeur, dont peuvent avoir besoin les DSA précédents de la chaîne. Ce type d'information est défini au 10.4.
- b) Le composant **result** – C'est une valeur de l'élément **operation.&Result** qui constitue le résultat retourné par le DSA ayant exécuté l'opération et qui est destiné à être communiqué dans le résultat au DUA demandeur. Cette information est telle que spécifiée à l'article approprié de la Rec. UIT-T X.511 | ISO/CEI 9594-3.

Si la demande échoue, une des erreurs énumérées dans l'ensemble **operation.&Errors** sera retournée, sauf qu'une erreur **dsaReferral** sera retournée à la place d'une erreur **referral**. L'ensemble des erreurs qui peuvent être signalées est celui qui est décrit pour l'opération correspondante dans la Rec. UIT-T X.511 | ISO/CEI 9594-3. L'erreur de type **dsaReferral** est décrite au 13.2.

12.2 Opération ChainedAbandon

L'opération **ChainedAbandon** est utilisée par un DSA pour indiquer à un autre DSA qu'il n'est plus intéressé par l'exécution d'une opération répartie qui a été invoquée antérieurement. Plusieurs raisons peuvent motiver cet abandon, comme les suivantes:

- l'opération qui a conduit le DSA à effectuer initialement un chaînage a elle-même été abandonnée ou a été interrompue implicitement par la terminaison d'une association;
- le DSA a obtenu les informations nécessaires d'une autre façon, par exemple de la part d'un DSA impliqué dans le multichainage parallèle et qui a répondu plus rapidement.

Un DSA n'est jamais obligé d'émettre un composant **ChainedAbandon**, ou d'abandonner une opération, si on lui demande de le faire.

Si le composant **ChainedAbandon** réussit effectivement à interrompre l'exécution d'une opération, un résultat sera retourné et l'opération concernée retournera une erreur **abandoned**. Si **ChainedAbandon** ne réussit pas à interrompre l'opération, il doit retourner lui-même une erreur de type **abandonFailed**.

13 Erreurs chaînées

13.1 Introduction

Dans le service abstrait de DSA, les erreurs qui peuvent être retournées sont, pour la plupart, les mêmes que celles du service abstrait d'Annuaire. Les exceptions sont que l'«erreur» **dsaReferral** (voir 13.2) est retournée au lieu de **Referral** et que les problèmes de service suivants ont la même syntaxe abstraite mais une sémantique différente.

- a) **invalidReference** – Le DSA qui retourne cette erreur a détecté une erreur dans les informations de connaissance du DSA appelant tel que spécifié dans l'argument de chaînage **referenceType**.
- b) **loopDetected** – Le DSA qui retourne cette erreur a détecté une boucle dans les informations de connaissance de l'Annuaire.

L'ordre de préséance des erreurs qui peuvent se produire est le même que dans le service d'Annuaire, tel que spécifié dans la Rec. UIT-T X.511 | ISO/CEI 9594-3.

13.2 Renvoi de référence de DSA

L'erreur **dsaReferral** est émise par un DSA qui, pour une raison quelconque, ne souhaite pas continuer l'exécution d'une opération en faisant suivre cette opération à un ou à plusieurs autres DSA par chaînage. Les circonstances dans lesquelles il peut retourner un renvoi de référence sont décrites au 8.3.

```

dsaReferral      ERROR      ::=  {
  PARAMETER      SET {
    reference      [0]      ContinuationReference,
    contextPrefix [1]      DistinguishedName OPTIONAL }
  CODE          id-errcode-dsaReferral }
    
```

Les divers paramètres ont le sens suivant:

- a) Le composant **ContinuationReference** contient les informations nécessaires à l'invocateur pour donner suite à une demande ultérieure appropriée, éventuellement vers un autre DSA. Ce type d'information est spécifié au 10.9.
- b) Si le composant **returnCrossRefs** de l'élément **ChainingArguments** pour cette opération avait la valeur **TRUE** et que le renvoi de référence soit fondé sur une référence subordonnée ou croisée, le paramètre **contextPrefix** peut être (le cas échéant) inclus. L'autorité administrative d'un DSA quelconque décidera quelles références d'informations de connaissance peuvent, le cas échéant, être retournées de cette façon (les autres pourront être, par exemple, confidentielles pour ce DSA).

SECTION 5 – PROCÉDURES DE FONCTIONNEMENT RÉPARTI

14 Introduction

14.1 Domaine d'application et limites

Cet article spécifie les procédures de fonctionnement réparti de l'Annuaire qui sont exécutées par des DSA. Chaque DSA exécute individuellement les procédures décrites ci-après; l'action collective de tous les DSA constitue l'intégralité des services fournis aux utilisateurs de l'Annuaire.

9594-4 : 1995 (F)**14.2 Conformité**

La description des procédures de DSA contenues dans la présente section est fondée sur les modèles des articles 8 et 9 de la Rec. UIT-T X.501 | ISO/CEI 9594-2 ainsi que des articles 7 et 8 de la présente Recommandation | Norme internationale. Les organigrammes et leurs descriptions textuelles correspondantes sont un moyen d'établir la correspondance entre une série de données d'entrée de DSA externes (protocole d'accès ou de système d'Annuaire) et une ou plusieurs sorties externes (à savoir un résultat, une erreur, un renvoi de référence ou des demandes chaînées) produites par le DSA en question, selon l'arbre des informations particulier de DSA détenu par ce DSA.

Il est probable que l'Annuaire sera réparti entre des DSA mis en œuvre conformément à différentes éditions des Spécifications d'Annuaire, comme les éditions 1988 et 1993. Le DUA lançant la demande n'aura pas connaissance de l'édition selon laquelle le (ou les) DSA répondant à la demande du DUA aura (auront) été mis en œuvre. Pour permettre le fonctionnement dans un environnement aussi hétérogène, un DSA doit être mis en œuvre conformément aux règles d'extensibilité définies dans l'article 7 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

La mise en œuvre d'un DSA doit être l'équivalent, sur le plan fonctionnel, du comportement externe spécifié par les procédures décrites ici. Les algorithmes utilisés par une mise en œuvre de DSA particulière en vue de déterminer la (ou les) sortie(s) correcte(s) à partir des entrées données et de l'arbre des informations de DSA, ne sont pas normalisés.

14.2.1 Interaction entre DSA édition 1988 et édition 1993

Si les opérations de modification s'étendent au-delà des limites du DSA (c'est-à-dire **AddEntry** avec **TargetSystem**, enlever ou renommer un préfixe de contexte), la présente Spécification d'Annuaire précise uniquement le comportement des DSA édition 1993. L'interaction entre deux DSA édition 1988 ou entre un DSA édition 1988 et un DSA édition 1993 n'est pas incluse dans le domaine d'application des présentes Spécifications d'Annuaire. Lorsque des DSA conformes à des éditions différentes sont en rattachement opérationnel hiérarchique, la connaissance par chacun de l'édition de ses homologues permettra de donner à l'utilisateur un message d'erreur homogène.

14.3 Modèle conceptuel

La complexité du fonctionnement réparti de l'Annuaire rend nécessaire une modélisation conceptuelle utilisant des techniques de description aussi bien narratives que graphiques. Toutefois, aucun des diagrammes narratifs ou graphiques ne saurait être interprété comme une description formelle du fonctionnement réparti de l'Annuaire.

14.4 Fonctionnement individuel et coopératif des DSA

Le modèle décrit le fonctionnement des DSA de deux points de vue séparés, dont la réunion donne une représentation opérationnelle complète de l'Annuaire.

- a) **Point de vue DSA** – Dans cette optique, l'ensemble des procédures qui gèrent l'Annuaire est décrit du point de vue d'un seul DSA. Cela permet de donner une spécification définitive de chaque procédure et de rendre intégralement compte de leurs relations et de la structure générale de commande. Les procédures des DSA du point de vue DSA sont décrites aux articles 16 à 22.
- b) **Point de vue opération** – Le point de vue DSA donne une description détaillée complète mais rend difficile la compréhension de la structure des opérations individuelles qui peuvent impliquer un traitement par plusieurs DSA. En conséquence, l'article 15 adopte un point de vue essentiellement axé sur les opérations pour présenter les phases de traitement applicables à chaque opération.

Afin de pouvoir gérer le fonctionnement réparti de l'Annuaire, chaque DSA doit effectuer les actions nécessaires pour réaliser l'objet de chaque opération et les actions supplémentaires nécessaires pour répartir cette réalisation entre plusieurs DSA. L'article 15 explore les différences entre ces deux catégories d'actions, qui sont spécifiées en détail aux articles 16 à 22.

14.5 Accords de coopération entre DSA

Tous les DSA qui sont en relation de subordonné/supérieur en raison des contextes de dénomination qu'ils détiennent ont entre eux des rattachements opérationnels hiérarchiques et/ou hiérarchiques non spécifiques, selon les types de référence de connaissance détenus par les DSA subordonnés.

On peut administrer les rattachements opérationnels hiérarchiques et hiérarchiques non spécifiques entre DSA au moyen des procédures des articles 24 et 25 de la présente Spécification d'Annuaire, ou par d'autres moyens (comme le téléphone).

Un DSA détenant des entrées qui relèvent de la zone administrative de son DSA supérieur doit administrer le sous-schéma et doit contrôler l'accès aux entrées selon les exigences de l'autorité administrative. La gestion des entrées à l'intérieur d'une zone administrative peut être assurée conformément à la Rec. UIT-T X.501 | ISO/CEI 9594-2, ou par des mécanismes locaux.

15 Comportement de l'Annuaire réparti

15.1 Exécution coopérative des opérations

Chaque DSA est doté des procédures capables d'exécuter complètement toutes les opérations d'Annuaire. Dans le cas où un DSA contient la DIB complète, toutes les opérations sont en fait complètement effectuées dans ce DSA. Dans le cas où la DIB est répartie entre plusieurs DSA, l'exécution d'une opération est normalement fragmentée, seule une partie de cette opération étant effectuée dans chacun des DSA coopérants, qui peuvent être nombreux.

En environnement réparti, le DSA voit normalement chaque opération comme un événement transitoire: l'opération est invoquée par un DUA ou par un autre DSA; le DSA effectue un traitement sur l'objet, puis le dirige vers un autre DSA pour traitement complémentaire.

Une autre manière de voir ce comportement est de considérer le traitement total subi par une opération au cours de son exécution par plusieurs DSA coopérants. Ce point de vue met en évidence les phases de traitement communes à toutes les opérations.

15.2 Phases de traitement d'une opération

Chaque opération d'Annuaire peut être considérée comme comprenant trois phases distinctes:

- a) la phase de *résolution du nom*, au cours de laquelle le nom de l'objet sur l'entrée duquel une opération particulière doit être effectuée, est utilisé pour localiser le DSA qui contient cette entrée;
- b) la *phase d'évaluation*, au cours de laquelle l'opération spécifiée par une demande particulière adressée à l'Annuaire (par exemple **Read**) est réellement effectuée;
- c) la *phase de fusion des résultats*, au cours de laquelle les résultats d'une opération spécifiée sont retournés au DUA demandeur. Si un mode de chaînage d'interactions a été choisi, la phase de fusion des résultats peut impliquer plusieurs DSA, dont chacun a chaîné la demande ou sous-demande initiale (définie au 15.3.1: Décomposition de demande) vers un autre DSA durant l'une ou l'autre des phases précédentes ou durant les deux.

Dans le cas des opérations Read, Compare, List, Search, Modify, Entry, ModifyDN et Remove Entry (lecture, comparaison, listage, recherche, modification, entrée, modification de nom distinctif et suppression d'entrée), la résolution du nom porte sur le nom d'objet fourni dans l'argument de l'opération. Dans le cas des opérations d'ajout d'entrée, l'entrée cible de la résolution de nom est l'entrée immédiatement supérieure à celle qui a été fournie dans l'argument de l'opération. Elle s'obtient aisément en supprimant le RDN final du nom fourni dans l'argument de l'opération. [Cela sera effectué au moyen de l'argument local **m** de la procédure FindDSE (trouver entrée spécifique de DSA) du 18.2.5.]

Une opération sur une entrée particulière peut initialement être dirigée vers n'importe quel DSA de l'Annuaire. Ce DSA utilise ses connaissances, éventuellement en conjonction avec d'autres DSA, pour traiter l'opération suivant les trois phases.

15.2.1 Phase de résolution du nom

La résolution du nom est le processus de mise en correspondance séquentielle de chaque RDN d'un nom prétendu avec un arc (ou nœud) du DIT, en commençant logiquement à la racine et en progressant vers le bas du DIT. Toutefois, comme le DIT est réparti de façon arbitraire entre de nombreux DSA, chaque DSA peut n'être capable d'exécuter qu'une fraction du processus de résolution du nom. Un DSA donné effectue sa part du processus de résolution du nom en parcourant son arbre des informations local de DSA. Ce processus est décrit à l'article 18 dans les diagrammes associés (Figures 9 à 12). Sur la base de son arbre des informations local de DSA, il peut savoir d'après les informations de connaissances qui y sont contenues, si la résolution peut être reprise en charge par un ou plusieurs autres DSA ou si le nom est erroné.

9594-4 : 1995 (F)**15.2.2 Phase d'évaluation**

Quand la phase de résolution du nom est terminée, l'opération requise proprement dite (par exemple **Read** ou **Search**) a lieu.

Les opérations qui impliquent une interrogation à simple entrée – Read et Compare – peuvent être exécutées entièrement à l'intérieur du DSA dans lequel cette entrée est située.

Les opérations qui impliquent une interrogation à entrées multiples – List et Search – doivent localiser des objets subordonnés de la cible, qui peuvent ou non résider dans le même DSA. Si tel n'est pas le cas, les opérations doivent être dirigées vers les DSA spécifiés dans les références subordonnées, subordonnées non spécifiques, fournisseur ou principales (selon le cas) pour exécuter le processus d'évaluation.

15.2.3 Phase de fusionnement des résultats

La phase de fusionnement des résultats commence au moment où quelques résultats de la phase d'évaluation sont disponibles.

Si l'opération ne concernait qu'une seule entrée, le résultat de cette opération peut simplement être retourné au DUA demandeur. Si l'opération concernait plusieurs entrées sur plusieurs DSA, les résultats doivent être combinés.

Les réponses permises, retournées à un demandeur après fusion des résultats, comprennent:

- a) un résultat complet de l'opération;
- b) un résultat qui n'est pas complet car certaines parties du DIT restent inexplorées (ne s'applique qu'au listage et à la recherche). Un tel *résultat partiel* peut incorporer des références de continuation concernant des parties inexplorées du DIT;
- c) une erreur (un renvoi de référence étant un cas spécial);
- d) et, si le demandeur était un DSA, un composant **ChainingResult**.

15.3 Gestion des opérations réparties

Des informations sont incluses dans l'argument de chaque opération dont l'exécution peut être demandée à un DSA: elles indiquent la progression de chaque opération au cours de sa traversée de divers DSA de l'Annuaire. Cela permet à chaque DSA d'exécuter la partie appropriée du traitement requis et d'enregistrer le fait que cette exécution a été effectuée, avant de diriger l'opération vers d'autres DSA.

Des procédures supplémentaires sont incorporées dans le DSA pour répartir physiquement les opérations et répondre à d'autres besoins résultant de leur répartition.

15.3.1 Décomposition de demande

La décomposition de la demande est un processus exécuté à l'intérieur d'un DSA avant qu'il entre en communication avec un ou plusieurs autres DSA. Une demande est décomposée en plusieurs sous-demandes, de manière telle que chacune de ces dernières accomplisse une partie de la tâche initiale. La décomposition de la demande peut être utilisée, par exemple dans l'opération recherche, après que l'entrée d'objet de base a été trouvée. Après la décomposition, chacune des sous-demandes peut être transmise par unichainage ou par multichainage à d'autres DSA, en vue de la poursuite de l'opération.

15.3.2 Réponse à une demande émanant d'un DSA

Un DSA qui reçoit une demande peut vérifier la progression de traitement de cette demande à l'aide du paramètre **operationProgress**, qui indiquera si l'opération est toujours en phase de résolution du nom ou si elle a atteint la phase d'évaluation et quelle partie de l'opération le DSA doit tenter de réaliser. Si le DSA ne peut pas satisfaire entièrement la demande, il doit soit communiquer l'opération par unichainage ou multichainage à un ou à plusieurs DSA qui peuvent aider à répondre à la demande ou retourner un renvoi de référence à un autre DSA ou mettre fin à la demande par un message d'erreur.

15.3.3 Exécution des opérations

Chaque DSA qui a lancé une opération ou qui l'a propagée vers un ou plusieurs autres DSA doit garder trace de l'existence de cette opération jusqu'à ce que chacun des autres DSA ait retourné un résultat ou une erreur, ou que le délai maximal de l'opération ait expiré. Cet impératif concerne toutes les opérations, tous les modes de propagation et toutes les phases de traitement. Il garantit la terminaison en bon ordre des opérations réparties qui se sont propagées dans l'Annuaire.

15.4 Traitement des boucles

Le DIT peut se trouver dans une situation entraînant un bouclage. Par exemple, un bouclage peut se produire au cours de la résolution d'un nom si le déréférencement d'un ou de plusieurs alias ramène la résolution à la même branche du DIT. Une erreur de configuration des références de connaissance est une autre cause possible de bouclage.

Dans le contexte d'une opération sur l'Annuaire particulière, une boucle se forme si, à un moment quelconque, il y a retour à l'état précédent, cet état étant défini par les composants suivants:

- le nom du DSA traitant actuellement l'opération;
- le nom de l'objet **targetObject**, tel que contenu dans l'argument de l'opération;
- le paramètre **operationProgress**, tel que contenu dans l'argument de l'opération et défini au 10.5.

Cela n'implique pas l'impossibilité qu'un même DSA traite plusieurs fois une même opération, mais le fait qu'un DSA ne traitera pas plusieurs fois la même opération dans le même état.

Les bouclages sont contrôlés au moyen de l'argument **traceInformation** défini au 10.6 qui enregistre la suite d'états qu'une opération donnée a traversés. Deux stratégies ont été définies pour déterminer si des boucles se sont formées ou vont se produire. Il s'agit de la détection des boucles et de la prévention des boucles; ces stratégies sont décrites respectivement aux 15.4.1 et 15.4.2.

La détection des boucles est obligatoire et la prévention des boucles est optionnelle.

15.4.1 Détection des boucles

A la réception d'une opération d'Annuaire, un DSA doit initialement valider l'opération pour faire en sorte qu'elle puisse se poursuivre. Une phase importante de la validation est la recherche des boucles, ce qui se fait en déterminant si l'état actuel de l'opération apparaît dans la suite d'états antérieurs enregistrés dans l'argument **traceInformation** relatif à cette opération. Cette étape du contrôle des bouclages constitue la détection des boucles.

15.4.2 Prévention des boucles

La prévention des boucles nécessite que le DSA détermine, immédiatement avant le transfert d'une opération à un autre DSA, dans le cadre d'une procédure de chaînage, si l'état consécutif de l'opération (qui est l'item **traceItem** que le DSA récepteur va ajouter à l'argument **traceInformation** quand il le recevra) apparaît dans la séquence des états précédemment enregistrés dans l'argument **traceInformation** pour l'opération entrante initiale.

En cas de réception ou de traitement de renvois de référence, on ne peut pas réaliser la détection et la prévention des boucles par simple examen de l'argument **traceInformation**. Si tel est le cas, chaque fois que le DSA agit sur un renvoi de référence, il doit enregistrer l'état de l'opération qui en résulte (c'est-à-dire le composant **traceItem** que le DSA récepteur va ajouter quand il recevra la demande) avec un enregistrement de la demande entrante. Avant d'agir sur un renvoi de référence ou de le retourner, le DSA doit vérifier dans cette liste s'il n'y a pas eu de demande identique envoyée au cours d'une précédente tentative de réaliser l'opération entrante.

15.5 Autres considérations relatives au fonctionnement réparti

15.5.1 Commandes de service

Certaines commandes de service nécessitent une prise en compte particulière, pour que l'opération soit traitée de la façon demandée.

- a) **chainingProhibited** – Un DSA consulte cette commande de service pour déterminer le mode de propagation d'une opération. Si elle est sélectionnée, le DSA utilise toujours le renvoi de référence. Si elle ne l'est pas, le DSA peut choisir d'utiliser le chaînage ou le renvoi de référence, selon ses possibilités.
- b) **timeLimit** – Un DSA doit tenir compte de cette commande de service pour garantir que le délai n'est pas dépassé à son niveau. Un DSA auquel un DUA demande d'exécuter une opération note en premier lieu le délai **timeLimit** donné par le DUA, en secondes, pour l'exécution de cette opération. Si un chaînage est nécessaire, le délai **timeLimit** est inclus dans l'argument de chaînage à communiquer à (ou aux) DSA suivant(s). Dans ce cas, la même valeur limite est utilisée pour chaque demande chaînée: c'est l'heure UTC à laquelle l'opération doit être achevée pour satisfaire la contrainte spécifiée initialement. Quand un DSA reçoit des arguments **ChainingArguments** où est spécifié une limite **timeLimit**, il doit respecter cette limite.

9594-4 : 1995 (F)

- c) **sizeLimit** – Un DSA doit tenir compte de cette commande de service pour garantir que la liste des résultats n'excède pas la taille spécifiée. La limite, telle qu'incluse dans l'argument commun de la demande initiale, est véhiculée telle quelle lorsque la demande est chaînée. Si une décomposition de la demande est requise, la même valeur est incluse dans l'argument à communiquer au DSA suivant: c'est-à-dire que la limite intégrale est utilisée pour chacune des sous-demandes. Quand les résultats sont retournés, le DSA demandeur résout les divers résultats et applique la limite à leur ensemble pour garantir que seul le nombre demandé est retourné. Si la limite a été dépassée, cela est indiqué dans la réponse.
- d) **priority** – Dans tous les modes de propagation, il incombe au DSA de s'assurer que le traitement des opérations est effectué dans un ordre permettant de se conformer à cette commande de service, si elle est présente.
- e) **localScope** – L'opération est limitée à un cadre défini au niveau local et aucun DSA ne peut propager la demande en dehors de ce cadre.
- f) **scopeOfReferral** – Si le DSA retourne un renvoi de référence ou un résultat partiel pour une opération **List** ou **Search**, les arguments **ContinuationReferences** doivent être imbriqués dans le cadre du renvoi demandé.

Toutes les autres commandes de service doivent être respectées, mais leur utilisation ne nécessite pas une prise en compte particulière dans l'environnement réparti.

15.5.2 Extensions

Si un DSA rencontre une opération étendue au cours de la phase de résolution du nom et s'il détermine que cette opération doit être chaînée à un ou à plusieurs DSA, il inclut dans l'opération chaînée, sans la modifier, toutes les extensions présentes.

NOTE – Une Autorité administrative peut juger approprié d'envoyer un message **serviceError** avec la cause **unwillingToPerform** si elle ne souhaite pas propager une extension.

Si un DSA rencontre une extension lors de la phase d'exécution du traitement, il existe deux possibilités. Si l'extension n'est pas critique, le DSA n'en tient pas compte. Si elle l'est, le DSA envoie un message **serviceError** avec la cause **unavailableCriticalExtension**. Une extension critique d'une opération à objets multiples peut se traduire à la fois par des résultats et par des erreurs de ce type. Un DSA qui fusionne de tels résultats et erreurs mettra au rebut ces erreurs de service et emploiera le composant **unavailableCriticalExtension** du qualificateur **PartialOutcomeQualifier** décrit dans la Rec. UIT-T X.511 | ISO/CEI 9594-3.

15.5.3 Déréférencement d'alias

Le déréférencement d'alias est le processus qui consiste à créer un nouveau nom d'objet cible, en remplaçant la partie relative au nom distinctif de l'entrée d'alias du nom d'objet cible initial par la valeur de l'attribut **AliasedEntryName** de l'entrée alias. Dans l'opération, le nom du composant **object** n'est pas affecté par le déréférencement d'alias.

15.6 Authentification des opérations réparties

Les utilisateurs de l'Annuaire ainsi que les autorités administratives assurant les services d'annuaire peuvent, à leur convenance, exiger l'authentification des opérations. Pour toutes les opérations d'annuaire, la nature de la procédure d'authentification dépend de la politique de sécurité en vigueur.

Il existe deux ensembles de procédures d'authentification qui satisfont toute une gamme de besoins d'authentification. Un de ces ensembles se compose des procédures assurées par l'opération de rattachement **Bind**, qui facilite l'authentification opérée entre deux entités d'application de l'annuaire afin d'établir une association. Les procédures de rattachement permettent toute une gamme d'échanges à des fins d'authentification, depuis le simple échange d'identités jusqu'à l'authentification renforcée.

En plus de l'authentification de l'entité homologue d'une association, comme celle qui est assurée par **Bind**, des procédures supplémentaires sont définies dans l'annuaire pour permettre l'authentification d'opérations individuelles. Deux ensembles distincts de procédures d'authentification sont ainsi définis. Le premier facilite les services d'authentification de l'expéditeur et concerne l'identification, par un DSA, de l'initiateur de la demande d'origine. Le second ensemble facilite les services d'authentification des résultats et concerne l'authentification, par un initiateur, de tous résultats retournés.

Deux procédures sont définies pour l'authentification de l'expéditeur, l'une fondée sur un simple échange d'identités, appelée **identity based authentication** et l'autre sur des techniques de signatures numériques, appelée **signature based authentication**. La première de ces procédures est de nature rudimentaire, puisque l'échange d'identités est fondé sur l'échange de noms spécifiques qui sont transmis en clair.

Pour l'authentification de résultats, une procédure unique (**results authentication**) est définie, fondée sur des techniques de signature numérique; du fait de la nature généralement complexe du collationnement des résultats, aucune procédure simple, fondée sur l'identité, n'est définie.

Ces procédures ne permettent pas de gérer l'authentification des réponses d'erreur.

Les services décrits ci-dessous doivent être considérés comme s'ajoutant à ceux qui sont assurés par le service Bind; les procédures de rattachement sont supposées avoir été exécutées avec succès avant l'authentification des opérations d'annuaire.

Les procédures que doit suivre un DSA pour authentifier l'expéditeur et les résultats d'authentification sont spécifiées à l'article 22.

16 L'aiguilleur d'opérations

L'aiguilleur d'opérations est la principale procédure de commande d'un DSA. Il fait passer chaque opération par les trois phases du traitement d'une demande. A cet effet, il dispose d'une série de procédures permettant de traiter complètement la demande, comme illustré à la Figure 6 ci-dessous.

16.1 Principes généraux

16.1.1 Procédures

Chacune des procédures utilisées par l'aiguilleur d'opérations est constituée d'une définition de son interface conceptuelle en termes de paramètres propres c'est-à-dire d'arguments, de résultats et d'erreurs, ainsi que d'une description des étapes de la procédure proprement dite. Le comportement de ces procédures est décrit au moyen d'organigrammes et de commentaires. Les symboles utilisés dans ces organigrammes ont la sémantique suivante (voir la Figure 7):

16.1.2 Emploi des structures de données communes

Toutes les procédures font appel à certaines structures de données qui sont disponibles au cours du traitement d'une opération guidée par l'aiguilleur d'opérations. Ces structures de données servent à coordonner la circulation des informations au sein de l'aiguilleur. La plupart de ces structures sont directement associées à l'argument de l'opération et au résultat qu'il y a lieu de créer pour l'opération. On se réfère aux composants de l'argument du résultat au moyen de leurs noms, donnés dans la définition ASN.1 correspondante (par exemple le composant **operationProgress** des arguments de chaînage). Si une de ces structures est composite, on peut se référer à un composant de la structure en question au moyen d'un composant composite (**compound.component**, par exemple **operationProgress.nameResolutionPhase**).

Les structures d'information suivantes sont définies au sein de l'aiguilleur d'opérations:

- **NRcontinuationList** – Liste de références de continuation créée pour être utilisée dans la procédure **NameResolutionContinuationReference**.
- **SRcontinuationList** – Liste de références de continuation créée pour être utilisée dans la procédure **List** ou **SearchContinuationReference**.
- **admPoints** – Liste de références à des entrées DSE de type «point d'administration» qui est établie au cours de la résolution du nom.

Par ailleurs, une procédure peut utiliser une série de variables définies sur place.

16.1.3 Erreurs

A chaque stade du traitement, une erreur peut être détectée au cours de l'exécution d'une sous-procédure. L'erreur identifiée dans cette sous-procédure est normalement retournée au demandeur sous forme d'erreur de protocole correspondante. Dans ce cas, l'aiguilleur d'opérations s'arrête immédiatement. En cas de réception de plusieurs erreurs, des procédures locales pourront choisir celle qui sera retournée.

Une procédure peut aussi choisir de traiter les erreurs (par exemple si le message **ServiceError busy** est retourné à une sous-demande de recherche chaînée) à certains points du traitement de l'opération. Dans ce cas l'exécution de la procédure se poursuit et aucune erreur n'est retournée au demandeur.

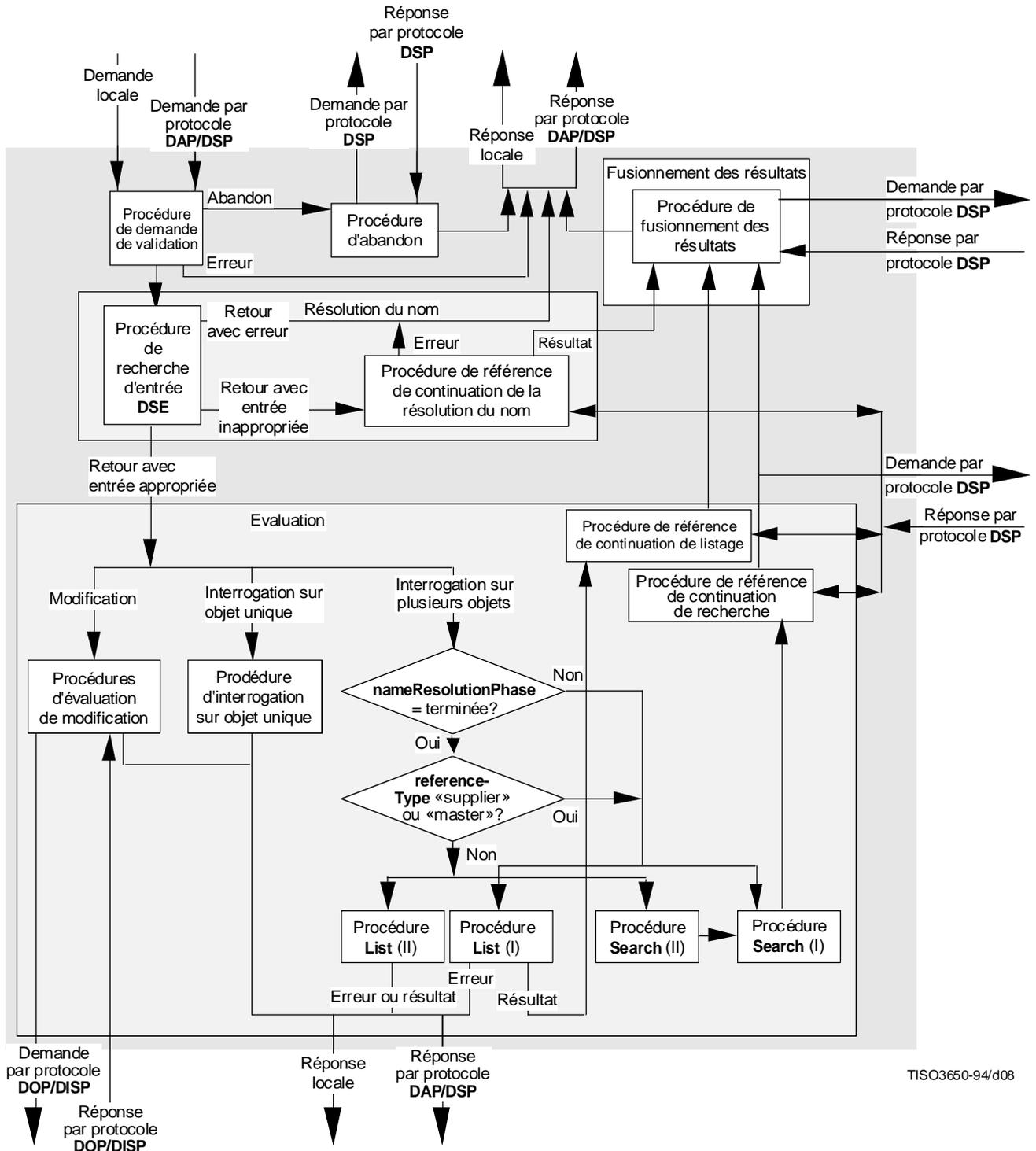


Figure 6 – Aiguilleur d'opérations

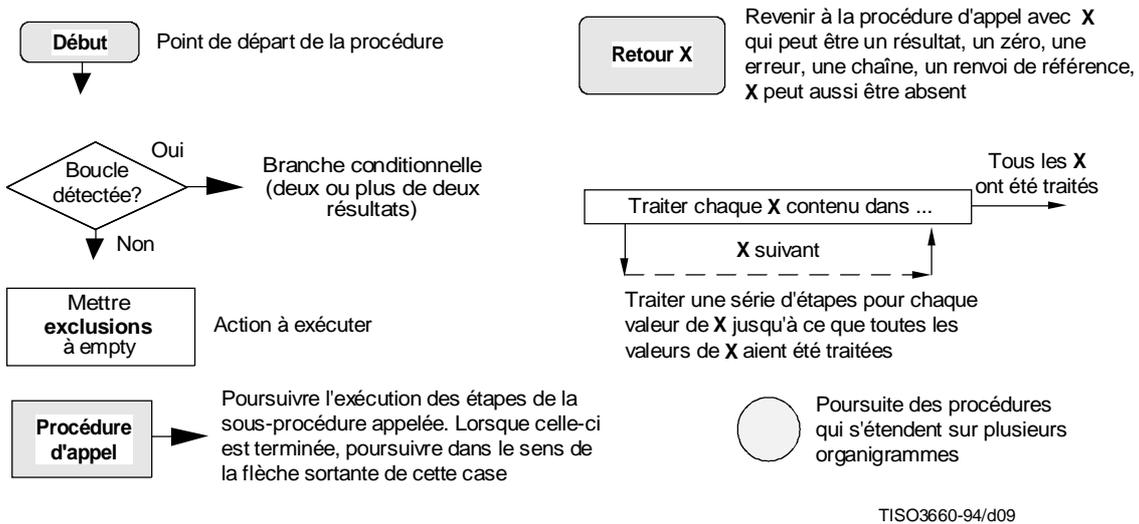


Figure 7 – Symboles utilisés dans les organigrammes

16.1.4 Événements asynchrones

Au cours du traitement d'une demande par l'aiguilleur d'opérations, plusieurs événements asynchrones peuvent survenir. Dans les paragraphes qui suivent, il est spécifié comment traiter un dépassement de limite de temps ou de limite de taille ou de limite administrative, une perte d'association et une demande d'abandon pour une opération qui est en cours de traitement. Le traitement de tous les autres événements asynchrones, tels que les décisions qui relèvent de la politique locale, etc. sont hors du domaine d'application de la présente Spécification.

16.1.4.1 Limite de temps

Une limite de temps (**timeLimit**), telle que spécifiée dans le composant **commonArguments**, peut arriver à expiration à n'importe quel moment de l'opération. Si cela se produit, c'est normalement une erreur **ServiceError** mentionnant la cause **timeLimitExceeded** qui est retournée au DUA ou DSA demandeur et l'aiguillage des opérations s'interrompt. Une autre solution consiste, pour la procédure, à choisir de traiter cet événement de manière différente (par exemple au cours du traitement d'une demande de recherche).

Si un DSA reçoit une demande d'un autre DSA quand la limite de temps est dépassée, il envoie une erreur **ServiceError** avec la cause **timeLimitExceeded** et ne poursuit pas le traitement de la demande.

Si le DSA a des (sous-)demandes en suspens quand la limite de temps (**timeLimit**) vient à expiration alors qu'aucun résultat n'est trouvé, il doit retourner au demandeur une erreur **ServiceError** avec la cause **timeLimitExceeded**.

Si le DSA a des sous-demandes en suspens quand la limite de temps expire et que des résultats ont été obtenus, il doit retourner au demandeur un résultat ayant le contenu suivant:

- tous les résultats recueillis jusqu'à l'expiration de la limite de temps (**timeLimit**);
- la composante **limitProblem** du paramètre de résultat **partialOutcomeQualifier**, mise à **timeLimitExceeded**;
- le composant **unexplored** du paramètre de résultat **partialOutcomeQualifier**, contenant une valeur de référence de continuation pour chaque ensemble de DSA auquel les sous-demandes ont été envoyées et dont le résultat n'est pas inclus dans le résultat retourné au demandeur, en plus des références de continuation concernant les DSA auxquels le DSA en question n'a pas tenté d'envoyer de sous-demandes.

16.1.4.2 Perte d'une association

En cas de perte d'association avec le demandeur, le DSA peut sur option, pour chaque (sous-)demande d'interrogation en suspens), envoyer une demande d'abandon, sauf si l'association avec le DSA en question a aussi été perdue. Toutes les réponses à de telles demandes d'abandon et toutes les (sous-)demandes en suspens seront rejetées.

9594-4 : 1995 (F)

En cas de perte de l'association avec une des sous-demandes chaînées en suspens alors que l'association avec le demandeur n'est pas perdue, le DSA peut (uniquement pour les opérations d'interrogation) essayer toute autre référence à un autre DSA qui a la capacité de traiter la demande chaînée (par exemple une référence à un DSA dupliqué, après perte de l'association avec le DSA maître). Si cela n'aboutit pas, le DSA agira de la manière suivante:

- 1) Si la variable **operationProgress.nameResolution** est mise à **notStarted** ou à **proceeding**, retourner au demandeur soit un message **ServiceError** indiquant la cause **unavailable**, ou une erreur de renvoi de référence dont la référence de continuation contiendra la série de DSA qui ont la capacité de poursuivre l'opération. Si l'on utilise des références subordonnées non spécifiques au cours de la phase de résolution du nom et que les associations en question ne soient pas toutes perdues, on pourra (le cas échéant) tenter d'opérer la résolution du nom sans les DSA avec lesquels les associations ont été perdues. Si cela échoue, retourner une erreur **ServiceError** avec la cause **unavailable** ou une erreur de renvoi de référence contenant l'ensemble complet de NSSR.
NOTE – Le type d'erreur à retourner relève d'un choix local.
- 2) Si la variable **operationProgress.nameResolution** est mise à **completed** et que la demande soit une opération d'interrogation à entrée unique, retourner au demandeur une erreur **ServiceError** ayant la cause **unavailable**.
- 3) Si la variable **operationProgress.nameResolution** est mise à **completed** et que la demande soit une interrogation à plusieurs entrées, le DSA ajoute une référence de continuation au composant **partialOutcomeQualifier.unexplored** du résultat de l'opération, avec une information **AccessPointInformation** identifiant l'ensemble des DSA qui sont capables de poursuivre l'opération, y compris tous ceux avec lesquels des associations ont été perdues.

16.1.4.3 Abandon de l'opération

Une demande d'abandon peut arriver pendant le traitement d'une opération. Dans ce cas, la procédure **Abandon** est invoquée au cours du traitement de la demande d'abandon pour l'opération qu'il y a lieu d'abandonner.

16.1.4.4 Limites administratives

L'administrateur de DSA local peut imposer des limites telles que la durée maximale de traitement d'une demande ou la longueur maximale des données à retourner. Si une de ces limites est dépassée, le DSA doit retourner un message **ServiceError** avec la cause **administrativeLimitExceeded**.

16.1.4.5 Limite de taille

Une limite de taille, telle que spécifiée dans le composant **CommonArguments**, peut être dépassée à un moment quelconque du traitement d'une opération **List** ou **Search**. Dans ce cas, un résultat partiel (choisi dans l'ensemble des résultats déjà recueillis) doit être retourné au demandeur avec l'erreur **limitProblem** mise à la cause **sizeLimitExceeded**. De plus, le composant **unexplored** peut être utilisé pour renvoyer les références de continuation de DSA qui n'ont pas été atteints. L'aiguilleur d'opérations est ensuite arrêté.

16.2 Procédure de l'aiguilleur d'opérations

La procédure utilisée par l'aiguilleur d'opérations pour traiter chaque demande reçue (soit par DAP, ou par DSP) est définie par les étapes ci-après. En raison du déréréférencement d'alias, cette procédure peut également s'appeler automatiquement (par demande locale). Dans ce cas, c'est une réponse locale (plutôt qu'une réponse par DAP ou DSP) qui sera retournée.

- 1) Valider les divers aspects des arguments d'opération (procédure **Request Validation**). Si une erreur est rencontrée au cours de la validation, retourner cette erreur localement ou par DAP/DSP.
- 2) Si l'opération reçue était une opération **Abandon**, appeler la procédure **Abandon** et retourner la réponse ensuite.
- 3) Résoudre le nom de l'objet cible en exécutant la procédure **Find DSE** (qui inclut les sous-procédures **Target Found** et **Target Not Found**). Si l'entrée demandée a été trouvée et qu'elle convienne (en vertu du choix des commandes de service, des arguments de chaînage et des décisions locales), poursuivre par la procédure **Evaluation Phase** à l'étape 6). Si, au cours de l'opération **Name Resolution**, on détecte une erreur, la retourner. Si l'entrée trouvée n'est pas jugée appropriée, passer à l'étape 4).
- 4) Appeler la procédure **Name Resolution Continuation Reference Procedure** pour traiter la liste des références de continuation telle qu'elle est enregistrée dans la liste **NRcontinuationList**. Pour traiter ces références de continuation, on peut émettre des demandes chaînées vers d'autres DSA (si les commandes de service ou les décisions locales le permettent).

En cas d'erreur, celle-ci est directement retournée soit localement soit via le DAP/DSP. Si la demande chaînée a produit un résultat, passer à l'étape 5).

- 5) La procédure **Result Merging Procedure** est appelée pour fusionner les résultats locaux avec les résultats chaînés reçus. Si les résultats chaînés contiennent des références de continuation imbriquées, elles ne pourront être résolues que si les commandes de service et la politique locale le permettent ou l'exigent.

Cela peut provoquer l'émission de demandes chaînées additionnelles (dont les résultats chaînés pourront également contenir des références de continuation).

Les résultats fusionnés sont retournés à l'appelant et le traitement de la demande est arrêté.

- 6) Si l'opération concerne une modification (**modification**), passer à l'étape 7.

Si l'opération concerne une interrogation à entrée unique (**single entry interrogation operation**), passer à l'étape 8.

Si l'opération concerne une interrogation à entrées multiples (**multiple entry interrogation operation**), passer à l'étape 9.
- 7) Dans l'exécution d'une procédure de type modification (**modification procedure**), il est parfois nécessaire d'établir des rattachements opérationnels, de les modifier ou d'y mettre fin, ou encore de mettre à jour les duplications miroirs par suite de l'exécution de l'opération. Que cela se fasse de manière synchronisée ou non avec l'exécution de l'opération initiale dépend des opérations de modification respectives (et de la politique locale). Un message de résultat local, de résultat par DAP/DSP ou d'erreur sera retourné à l'appelant.
- 8) Le résultat d'une opération de type **single entry interrogation operation** sera directement retourné à l'appelant sous forme d'un résultat local ou d'un résultat par DAP/DSP.
- 9) Si l'opération est de type **multiple entry interrogation operation**, vérifier l'élément **nameResolutionPhase** de l'opération. S'il n'est pas à la valeur **completed**, appeler la procédure **List(I)** ou **Search(I)**, ou la procédure **List(II)** ou **Search(II)**, selon le cas.
- 10) Le résultat d'un appel à la procédure **List(II)** (résultat ou erreur) et le résultat d'un appel à la procédure **List(I)** (si ce résultat est une erreur) peut être directement retourné à l'appelant (en tant que résultat local ou résultat par DAP/DSP).

Si la procédure appelée était la procédure **List(I)**, le résultat peut contenir des références de continuation qui doivent être déréférencées (selon les commandes de service et la politique locale). Cela peut se traduire par l'envoi d'opérations List chaînées aux DSA respectifs. Pour fusionner les résultats, passer à l'étape 5) avec l'appel à la procédure **Result Merging Procedure**.
- 11) Si l'opération était une opération Search, toute référence de continuation est résolue par la procédure **Search Continuation Reference Procedure** (si elle est nécessaire et permise). Cela peut entraîner l'envoi de demandes Search chaînées aux DSA respectifs. La procédure **Result Merging Procedure** (étape 5) est appelée pour fusionner les résultats des recherches et pour déréférencer si possible les références de continuation qui y sont éventuellement contenues.

16.3 Vue d'ensemble des procédures

Le présent article est une vue d'ensemble des fonctionnalités de base des procédures employées par l'aiguilleur d'opérations et qui sont définies aux articles 17 à 22.

16.3.1 Procédure de validation de demande (Request Validation procedure)

Cette procédure, décrite à l'article 17, est appelée afin d'exécuter les vérifications relatives aux bouclages, aux limites et à la sécurité, avant d'effectuer la résolution locale du nom. Elle établit également les réglages par défaut des paramètres de l'argument **chainingArgument** qui ne sont pas fournis par le DAP si la demande émane d'un DUA. Par ailleurs, cette procédure isole toute demande **Abandon** et en avise l'aiguilleur d'opérations.

16.3.2 Procédure d'abandon (Abandon procedure)

Cette procédure, décrite au 20.5, cherche l'opération qu'il y a lieu d'abandonner et tente d'y mettre fin. Au cas où il y aurait une sous-demande en suspens, des opérations **Chained Abandon** peuvent être envoyées à leur suite. La procédure retourne à l'appelant soit un **Null Result**, ou une indication d'erreur (par exemple une erreur **AbandonError** avec la cause **tooLate**).

9594-4 : 1995 (F)

16.3.3 Procédure de recherche de DSE (Find DSE procedure)

Cette procédure, décrite aux 18.2 et 18.3, adapte les composants du nom de l'objet cible aux DSE détenues localement afin de résoudre le nom de l'objet cible. En cas de rencontre d'une DSE alias, cet alias est déréférencé (si cela est permis) et la procédure est relancée pour résoudre le nouveau nom.

Si la cible n'est pas trouvée, le traitement se poursuit par la sous-procédure **Target Not Found sub-procedure**. Si la cible est trouvée, le traitement se poursuit par la sous-procédure **Target Found sub-procedure**.

NOTE – Les sous-procédures **Target Not Found** et **Target Found** sont des continuations de la procédure **Find DSE**.

La procédure peut aboutir à diverses erreurs; dans ce cas l'erreur de protocole associée est retournée au demandeur et il est mis fin à l'aiguillage des opérations.

16.3.3.1 Sous-procédure de cible non trouvée (Target Not Found sub-procedure)

Cette procédure, décrite au 18.3.2, effectue une évaluation des DSE intermédiaires locales et crée un ensemble de références **continuationReferences** dans la liste **NRcontinuationList**, sur la base de l'ensemble des références de connaissances qui ont été détectées au cours de la procédure **Find DSE**. Cet ensemble de références est ensuite traité par la procédure de référence **Name Resolution Continuation Reference**.

Cette procédure peut aboutir à diverses erreurs; si c'est le cas, l'erreur associée est retournée au demandeur et il est mis fin à l'aiguillage des opérations.

16.3.3.2 Sous-procédure de cible trouvée (Target Found sub-procedure)

Cette procédure, définie au 18.3.3, vérifie si la DSE qui a été trouvée convient à l'opération demandée, c'est-à-dire s'il s'agit d'une information dupliquée. Cela peut inclure le contrôle de l'utilité de l'ensemble du sous-arbre des informations miroirs au-dessous de l'objet cible dans le cas d'une opération sur plusieurs objets (par exemple une recherche sous-arborescente).

Si l'entrée qui a été localisée convient, on invoque la procédure d'évaluation de l'opération appropriée. Dans le cas contraire, une référence **continuationReference** pointant vers le fournisseur (ou maître) de l'information est créée dans la liste **NRcontinuationList** et c'est la procédure de référence **Name Resolution Continuation Reference** qui est alors invoquée.

16.3.4 Procédure d'interrogation à entrée unique (Single entry interrogation procedure)

Cette procédure, décrite au 19.2, est invoquée pour exécuter finalement les opérations qui n'interviennent que sur une entrée unique, à savoir **Read** et **Compare**. Lorsqu'elle est terminée, une réponse (résultat ou erreur) créée par la procédure est retournée au DSA/DUA demandeur.

16.3.5 Procédures de modification (Modification procedures)

Ces procédures, décrites au 19.1, sont exécutées pour traiter les opérations de modification, à savoir **AddEntry**, **RemoveEntry**, **ModifyEntry** et **ModifyDN**. Cela se fait par l'exécution d'une sous-procédure spécifique, définie pour chacune de ces opérations. Au cours de ces procédures (ou après), des demandes par DOP et DISP peuvent être émises vers d'autres DSA. Après exécution normale, un résultat (créé par les sous-procédures) est retourné au DSA/DUA demandeur.

16.3.6 Procédures d'interrogation à entrées multiples (Multiple entry interrogation procedures)

Ces procédures, décrites au 19.3, sont exécutées pour traiter des opérations qui interviennent sur plusieurs entrées pouvant (le cas échéant) être situées dans le même DSA. Cela se fait en exécutant des procédures spécifiques, définies pour chacune des opérations **Search** et **List** afin de réaliser la décomposition de la demande. Ces procédures créent un résultat local de l'évaluation de l'opération et établissent (le cas échéant) un ensemble de références de continuation dans la liste **SRcontinuationList**. Si celle-ci est vide à la fin de la procédure, le résultat créé est directement retourné au DSA/DUA demandeur. A défaut, ces références de continuation sont traitées par invocation de la procédure **List** ou **Search Continuation Reference** selon le type d'opération dont il s'agit.

16.3.7 Procédure de référence de continuation de résolution de nom (Name resolution Continuation Reference procedure)

Cette procédure, décrite au 20.4.1, traite les références de continuation contenues dans la liste **NRcontinuationList** créée au cours de la phase de résolution du nom. Ces références de continuation sont utilisées pour émettre des sous-demandes chaînées ou sont retournées dans une erreur de renvoi de référence. En cas de chaînage, les résultats ou erreurs émanant de la demande chaînée sont retournés pour traitement ultérieur par la procédure de fusionnement des résultats (**Result Merging**).

16.3.8 Procédure de référence de continuation des opérations de listage et de recherche (List and Search Continuation Reference procedure)

Ces procédures, décrites aux 20.4.3 et 20.4.4, traitent les références de continuation contenues dans la liste **SRcontinuationList** créée par les procédures d'interrogation à entrées multiples, soit pour les résoudre en émettant des sous-demandes chaînées ou en créant une ou plusieurs références de continuation au sein de la variable **partialOutcomeQualifier.unexplored**. Quand les résultats ou erreurs pour toutes les sous-demandes en suspens ont été obtenus, ils sont retournés pour suite de traitement par la procédure de fusionnement des résultats (**Result Merging**).

16.3.9 Procédure de fusionnement des résultats (Result Merging procedure)

Cette procédure, décrite à l'article 22, a pour effet d'examiner le résultat d'une demande chaînée ou de combiner les résultats des opérations locales avec les résultats reçus des sous-demandes chaînées. Si une sous-demande a retourné une erreur, cette procédure détermine la manière dont cette erreur doit être traitée.

Si d'éventuelles références de continuation sont laissées dans le résultat, elles seront déréférencées (si cela est permis par la politique locale et si les commandes de service le requièrent) par la procédure **Name Resolution, List** ou **Search Continuation Reference**, selon le cas. Les valeurs en double seront retirées du résultat si celui-ci est non signé.

Le résultat fusionné (c'est-à-dire tous les résultats fusionnés et toutes les références de continuation non résolues) est retourné au DUA/DSA demandeur.

17 Validation de demande

17.1 Introduction

Il s'agit d'une procédure qui représente le point d'entrée de l'aiguilleur d'opérations pour les entrées issues des DUA et DSA et qui prépare de telles entrées en vue du traitement de résolution du nom. La fonction de cette procédure est de détecter les opérations d'abandon, d'effectuer les contrôles de sécurité, d'ajuster les entrées reçues des DUA afin qu'elles puissent être traitées de la même manière que les entrées reçues des DSA, de vérifier les arguments de la demande de validité syntaxique et sémantique, de détecter les boucles et d'effectuer divers autres contrôles. Le déroulement de la validation de la demande est illustré à la Figure 8.

17.2 Paramètres de la procédure

17.2.1 Arguments

Les arguments d'entrée à la validation de demande sont les composants **ChainingArguments** (sauf dans le cas des opérations de type **ChainedAbandon**) si la demande émane d'un DSA et si l'argument est émis par l'expéditeur de la demande.

17.2.2 Résultats

Les cinq résultats possibles restitués par la procédure de validation de demande sont les suivants:

- a) si le contrôle de sécurité est négatif, une erreur est retournée au demandeur;
- b) si l'entrée est une opération **Abandon** ou **ChainedAbandon**, le résultat est l'argument de l'opération;
- c) si les arguments de la demande sont invalides, une erreur est retournée au demandeur. Compte tenu de la politique locale, le DSA peut toutefois retourner un message **ServiceError** ou **SecurityError**;
- d) en cas de détection d'une boucle, c'est une erreur de type **ServiceError** avec la cause **loopDetected** qui sera retournée au demandeur;
- e) si, pour des problèmes de ressources ou des considérations de politique, le DSA ne peut ou ne souhaite pas effectuer l'opération, une erreur de type **ServiceError** (avec la cause **busy**, **unavailable** ou **unwillingToPerform**) est retournée au demandeur;
- f) dans tous les autres cas l'entrée validée, transformée par adjonction des arguments **ChainingArguments** si elle émane d'un DUA ou par la mise à jour de l'information **ChainingArguments.traceInformation** si elle émane d'un DSA, constitue le résultat de la procédure et devient ensuite l'entrée pour la résolution du nom.

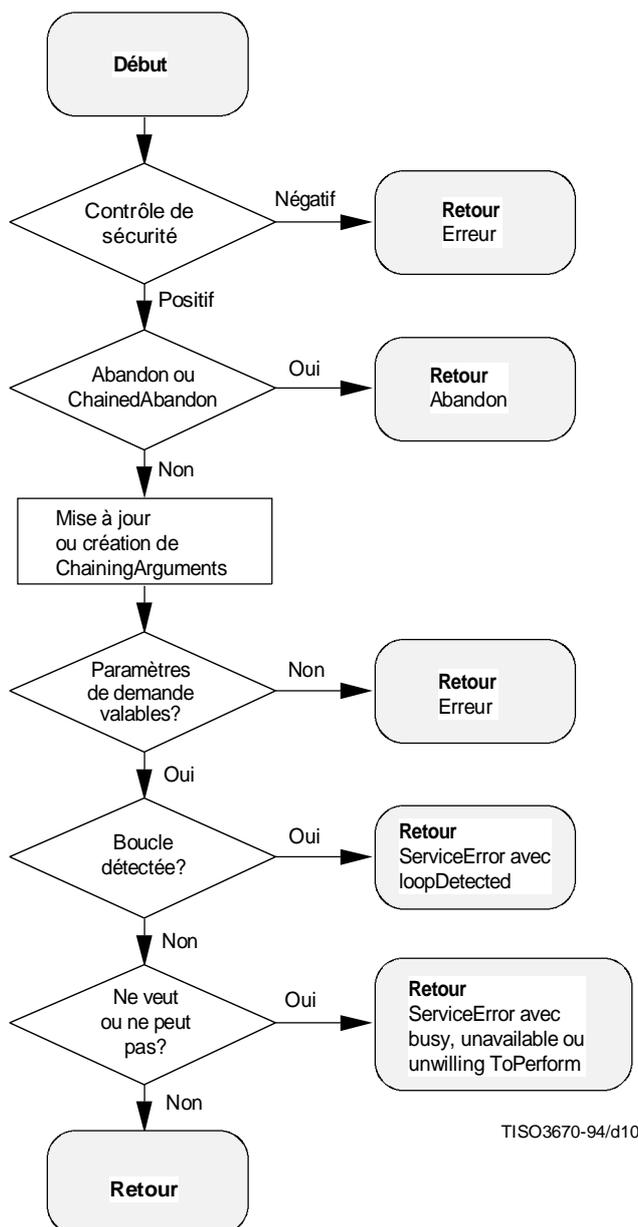


Figure 8 – Procédure de validation de demande

17.3 Définition de la procédure

Le contrôle de sécurité décrit au 17.3.2 est effectué. Cela peut se traduire par le retour d'une erreur et par l'arrêt de l'aiguillage des opérations.

Si l'entrée est une opération **Abandon** ou **ChainedAbandon**, seules les étapes du 17.3.1 sont ensuite effectuées; à défaut, ce sont celles des 17.3.3 à 17.3.5. Le 17.3.5 décrit la procédure de détection des boucles, qui peut aboutir au retour d'une erreur et à l'arrêt de l'aiguillage des opérations.

Ensuite les contrôles du 17.3.6 sont effectués. Ceux-ci peuvent aboutir au retour d'une erreur et à l'arrêt de l'aiguillage des opérations.

Si les contrôles des 17.3.2 à 17.3.6 n'aboutissent pas à l'arrêt de l'aiguillage des opérations, ce sont les opérations du 17.3.7 qui sont effectuées et la procédure se termine par le transfert de son résultat à la procédure de résolution du nom.

17.3.1 Traitement d'un abandon

L'argument d'une opération **Abandon** ou **ChainedAbandon** est transmis à la procédure **Abandon** (voir le 20.5), en vue du traitement de cette demande.

17.3.2 Contrôles de sécurité

Si l'argument relatif à l'opération est signé, cette signature peut être contrôlée. Si celle-ci n'est pas valable ou absente alors qu'elle devrait être présente, une erreur peut être retournée au demandeur. Une autre possibilité consiste pour le DSA à effectuer une autre action, définie localement.

17.3.3 Préparation à l'entrée

17.3.3.1 Demande émanant d'un DUA

Si l'opération émane d'un DUA, une valeur **ChainingArguments** est créée de la manière suivante.

- a) Le composant **ChainingArguments.originator** est défini comme indiqué au 10.3.
- b) Le composant **ChainingArguments.traceInformation** est mis à une séquence contenant une valeur de **TraceItem** unique. Cette valeur est construite de la manière suivante: le composant **TraceItem.dsa** prend le nom du DSA exécutant la validation de la demande, le composant **TraceItem.targetObject** est omis. Le paramètre **TraceItem.operationProgress** est mis à la valeur entrante.
- c) Si la commande de service de l'opération spécifie une limite de temps (temps disponible, en secondes, pour l'exécution de l'opération), le composant **ChainingArguments.timeLimit** est mis à l'heure UTC à laquelle l'opération devra être terminée pour répondre à la limite de temps spécifiée par l'utilisateur.
- d) Les composants **ChainingArguments.AuthenticationLevel** et **ChainingArgument.UniqueID** sont sélectionnés conformément à la politique de sécurité locale.
- e) Les composants facultatifs restants de **ChainingArguments** sont omis, les valeurs par défaut étant adoptées lorsqu'elles sont spécifiées.

17.3.3.2 Demande émanant d'un DSA

Si la demande émane d'un DSA, le composant **ChainingArguments.traceInformation** est mis à jour par l'adjonction d'une valeur à la fin de la séquence **TraceItem**. Cette valeur est construite de la manière suivante:

- a) **TraceItem.dsa** est mis au nom du DSA exécutant la validation de la demande.
- b) **TraceItem.targetObject** est mis à la valeur de **ChainingArguments.targetObject** à moins que **object** (ou **baseObject** dans le cas d'une recherche) de l'argument de recherche ne soit identique à **ChainingArguments.targetObject**, en quel cas **TraceItem.targetObject** est omis.
- c) **TraceItem.operationProgress** est mis à la valeur de **ChainingArguments.operationProgress**.

17.3.4 Assertion de validité

Cette opération est contrôlée au plan de la validité syntaxique et sémantique de ses arguments conformément aux règles contenues dans les articles définissant chaque opération (il convient par exemple de vérifier que le numéro **nextRDNTToBeResolved** ne donne pas un nombre dépassant le nombre de RDN du **targetObject**). S'il apparaît que la demande contient des arguments invalides, il est mis fin à l'opération et une erreur est retournée à l'utilisateur en fonction du type de non-validité constatée.

17.3.5 Détection de boucle

Si deux valeurs quelconques de **TraceItem** de **ChainingArguments.traceInformation** (comme préparé au 17.3.3) sont identiques, le traitement de l'opération est revenu à un état précédent, c'est-à-dire qu'une boucle a été détectée. Dans ce cas, une erreur **ServiceError** (avec la cause **loopDetected**) est retournée au demandeur et il est mis fin à l'aiguillage des opérations.

9594-4 : 1995 (F)

17.3.6 Incapacité ou refus d'exécution

La procédure de validation de demande peut évaluer les ressources disponibles et déterminer que l'opération ne peut être effectuée. Elle peut également déterminer, sur la base de considérations politiques, qu'il n'y a pas lieu d'effectuer l'opération. Dans ces cas, une erreur **ServiceError** (avec la cause **busy**, **unavailable** ou **unwillingToPerform**) peut être retournée au demandeur et on arrête l'aiguillage des opérations.

17.3.7 Traitement à la sortie

Dans la phase finale de la procédure de validation de demande, l'entrée validée, transformée par adjonction de **ChainingArguments** si elle émane d'un DUA ou par la mise à jour de **ChainingArguments.traceInformation** si elle émane d'un DSA, est renvoyée et utilisée comme entrée pour la procédure de résolution du nom.

18 Résolution du nom**18.1 Introduction**

Le présent article contient la description de la procédure de résolution du nom, de ses arguments, de ses résultats et de ses éventuelles conditions d'erreur. Ainsi qu'il est montré à la Figure 6 (Aiguilleur d'opérations), la procédure de résolution du nom comprend deux parties:

- la procédure **Find DSE**;
- la procédure **Name Resolution Continuation Reference**.

La procédure **Find DSE** est décrite au moyen de trois organigrammes, à savoir **Find DSE**, **Target Found** et **Target Not Found**. La procédure **Find DSE** établit la correspondance entre le nom d'entrée cible et les DSE enregistrées sur place, composant par composant. Si l'entrée cible est trouvée sur place, la procédure **Find DSE** est suivie de la sous-procédure **Target Found**, qui à son tour appelle la procédure **Check Suitability** afin d'évaluer la DSE venant d'être trouvée. Si l'entrée cible n'est pas trouvée sur place, la procédure **Find DSE** est suivie de la sous-procédure **Target Not Found** qui prépare une ou plusieurs références de continuation à ajouter à la liste **NRcontinuationList** en vue de l'aiguillage par la procédure **Name Resolution Continuation Reference**.

18.2 Paramètres de la procédure Find DSE**18.2.1 Arguments**

La procédure utilise les arguments suivants:

- a) **chainingArguments.aliasDereferenced**;
- b) **chainingArguments.aliasedRDNs**;
- c) **chainingArguments.excludeShadows**;
- d) **chainingArguments.nameResolveOnMaster**;
- e) **chainingArguments.operationProgress (nameResolutionPhase,nextRDNTToBeResolved)**;
- f) **chainingArguments.referenceType**;
- g) **chainingArguments.targetObject**;
- h) **commonArguments.serviceControls.copyShallDo**;
- i) **commonArguments.serviceControls.dontDereferenceAliases**;
- j) **commonArguments.serviceControls.dontUseCopy**.

NOTE – Si aucune valeur précise n'existe, on utilisera les valeurs par défaut ou implicites, comme spécifié au 10.3.

18.2.2 Résultats

Deux cas de réussite de la procédure **Find DSE** sont possibles (désignés respectivement **entry suitable** et **entry unsuitable**):

Le premier retourne (en provenance de la sous-procédure **Target Not Found**) une ou plusieurs références de continuation contenue(s) dans **NRcontinuationList** qui est(ont) ensuite transmise(s) à la procédure **Name Resolution Continuation Reference** afin de poursuivre la phase de résolution du nom.

Le second retourne (en provenance de la sous-procédure **Target Found**) une (référence à une) DSE qui est transmise à l'une des procédures d'évaluation.

18.2.3 Erreurs

Les erreurs suivantes peuvent être retournées:

- a) **ServiceError: unableToProceed, invalidReference, unavailableCriticalExtension;**
- b) **NameError: noSuchObject, aliasDereferencingProblem.**

18.2.4 Variables globales

La procédure fait appel aux variables globales suivantes:

- **NRcontinuationList**: liste d'enregistrement de la ou des références de continuation nécessaires à la poursuite de la résolution du nom dans la procédure **Name Resolution Continuation Reference**.

18.2.5 Variables locales et partagées

La procédure utilise les variables locales suivantes:

- a) **i** Indice utilisé pour identifier le composant du nom cible sur lequel s'effectue le traitement;
- b) **m** Longueur du nom d'objet cible à utiliser pour la résolution du nom. Pour les opérations qui résolvent le nom par rapport à l'entrée parente (c'est-à-dire Add Entry, la valeur **m** est mise au (nombre de RDN contenus dans l'objet cible – 1). Pour toutes les autres opérations la valeur **m** est mise au nombre de RDN contenus dans l'objet cible.
- c) **lastEntryFound** Indice tel que la DSE (lastEntryFound) soit la dernière entrée DSE de type **entry**;
- d) **lastCP** Indice tel que la DSE (lastCP) soit le dernier préfixe de contexte dupliqué en miroir qui a été rencontré;
- e) **candidateRefs** Ensemble de références de continuation.

La variable partagée **admPoints** (définie dans l'aiguilleur d'opérations) est également utilisée. Pour des raisons de commodité, la composante **i** du nom de l'objet cible est représentée par **N(i)**.

18.3 Procédures

18.3.1 Procédure Find DSE

Voir la Figure 9.

Cette procédure cherche à résoudre le nom d'objet localement.

- 1) Initialiser les variables locales **lastEntryFound** et **lastCP** à 0; **admPoints** et **candidateRefs** à un ensemble vide et **i** à 0.
- 2) Comparer **i** et **m**. S'ils ne sont pas égaux, passer à l'étape 7).
- 3) S'ils sont égaux, vérifier si la phase **nameResolutionPhase** est à **completed**. Dans la négative, passer à la sous-procédure **Target Not Found**.
- 4) Si la phase de résolution du nom est déjà terminée, vérifier si une quelconque subordonnée de la DSE(**i**) est un préfixe de contexte (de type **cp**).

Si une ou plusieurs subordonnées le sont, retourner **entry suitable**.

NOTE 1 – Ce cas s'applique à des sous-demandes **List(II)** et **Search(II)**.

Si aucune ne l'est, poursuivre avec la sous-procédure **Target Not Found**.

- 5) Tenter de trouver une concordance entre le (**i + 1**)ème composant du nom de l'objet cible et le nom d'une subordonnée de la dernière DSE avec laquelle il y avait concordance. Dans le cas de **i = 0**, tenter d'établir une correspondance entre l'une des DSE immédiatement subordonnées et la DSE racine. Si l'on ne peut trouver de concordance, passer à la sous-procédure **Target Not Found**. S'il y a concordance, incrémenter **i** et enregistrer la DSE, pour laquelle il y a concordance, comme étant le **i**-ème élément dans le vecteur des DSE trouvées.

9594-4 : 1995 (F)

- 6) Si i égale **nextRDNTToBeResolved**, vérifier si les deux conditions suivantes sont remplies:
- la variable **nameResolveOnMaster** de l'argument de chaînage est à **True**;
 - la DSE(i) n'est pas une entrée principale (c'est-à-dire de type duplication miroir).

Si les deux conditions sont remplies, retourner une erreur de service avec **unableToProceed**.

NOTE 2 – Cela indique l'utilisation de **nameResolveOnMaster** pour éviter les trajets multiples vers le même objet cible.

- 7) Vérifier tous les bits de type DSE de DSE(i). Pour chaque bit correspondant à un type, un certain traitement est sans doute requis. L'action à entreprendre pour chaque type trouvé est donnée ci-dessous:

- si les bits **cp** et **shadow** sont tous deux sélectionnés, mémoriser l'indice i dans le préfixe **lastCP**.
- Si le bit **admPoint** est sélectionné, vérifier l'attribut opérationnel **administrativeRole**. Si c'est le début d'une zone administrative autonome, vider la liste des points **admPoints**. Si c'est le début d'une ou de plusieurs zone(s) administrative(s) spécifique(s), vérifier la liste des points administratifs et éliminer tous les points existants qui ne sont plus applicables (à savoir que leur rôle a été remplacé par celui des nouveaux points administratifs). Enregistrer dans la liste les DSE(i).
- Si l'un des bits **subr**, **xr** ou **immSupr** est sélectionné, produire une référence de continuation utilisant l'attribut **SpecificKnowledge** avec **operationProgress.nameResolutionPhase** mis à **proceeding**, **nextRDNTToBeResolved** mis à i et **accessPoints** et **referenceType** mis comme il convient. Ajouter la référence de continuation à la liste des références de continuation dans **candidateRefs**.
- Si le bit **entry** est sélectionné, essayer i égal à m (et pour cette raison le nom de l'objet cible aura une concordance complète). Si i n'est pas égal à m , mémoriser l'entrée trouvée en mettant **lastEntryFound** à i et continuer le traitement des bits de type de la DSE(i). Si i et m sont égaux, passer à l'étape 10).
- Si le bit **subentry** est sélectionné, faire l'essai de i égal à m (dans ce cas le nom de l'objet cible aura une concordance complète). S'ils sont égaux, poursuivre avec la sous-procédure **Target Found**; s'ils ne le sont pas, retourner une erreur **NameError** avec la cause **noSuchObject**.
- Si le bit **alias** est sélectionné, vérifier si le composant **dontDereferenceAliases** est sélectionné.

S'il ne l'est pas, l'alias peut être déréférencé. Pour cela, mettre **chainingArguments.aliasDereferenced** à **TRUE**, **nameResolutionPhase** à **notStarted**, le nom de l'objet cible à **aliasedEntryName** tel que fourni dans l'entrée alias concaténée avec les composants non concordants qui restent du nom de l'objet cible précédent (c'est-à-dire concaténer avec les ($i + 1$)ème au m ème composants du nom de l'objet cible précédent). Les DSA édition 1993 mettront les **aliasedRDN** à **1**, alors que les DSA édition 1988 mettront les **aliasRDN** à la valeur i . Relancer la résolution du nom en passant à l'étape 9).

Si **dontDereferenceAliases** est sélectionné, l'alias ne peut être déréférencé. Vérifier si le nom de l'objet cible a été complètement traité en comparant i et m en ce qui concerne leur égalité. S'ils sont égaux (le nom aura alors une concordance complète), poursuivre au moyen de la sous-procédure **Target Found**. S'ils ne sont pas égaux (et que le nom n'ait pas de concordance complète), retourner **NameError** avec la cause **aliasDereferencingProblem**.

- Pour tous les autres types de DSE possibles, aucune action n'est requise. Marquer de façon interne que ce type de DSE est traité et poursuivre le traitement des bits de type de DSE(i) qui ne sont toujours pas traités.
 - Si tous les bits de type de DSE(i) sont traités, passer à l'étape 2).
- 8) Vérifier si la phase de résolution du nom est déjà terminée. Si elle ne l'est pas, passer à la sous-procédure **Target Found**.
- 9) Si la phase de résolution du nom est terminée, vérifier que le type **referenceType** utilisé est **supplier** ou **master**; si tel est le cas, poursuivre avec la sous-procédure **Target Found**.

NOTE 3 – Cela se fait pour les sous-demandes chaînées au fournisseur.

- 10) Si tel n'est pas le cas, vérifier si l'une quelconque des DSE subordonnées à la DSE(i) est un préfixe de contexte (et, pour cette raison, est de type **cp**). Dans l'affirmative (une ou plusieurs subordonnées), retourner **entry suitable**. Si aucune des entrées subordonnées n'est du type préfixe de contexte, retourner une erreur de type **ServiceError** avec la cause **invalidReference**.

NOTE 4 – Ce cas s'applique à des sous-demandes **List (II)** et **Search (II)**.

9594-4 : 1995 (F)

18.3.2 Sous-procédure Target Not Found

Voir la Figure 10.

Cette sous-procédure est appelée quand le nom de l'objet cible n'est pas trouvé dans le DSA local. La sous-procédure détermine le meilleur type de référence de connaissance qu'il convient d'utiliser pour continuer la résolution du nom à moins qu'une erreur ne soit détectée; dans ce cas une erreur est retournée.

- 1) En poursuivant depuis la procédure **Find DSE**, il convient de faire la distinction entre les trois possibilités de la phase de résolution du nom.
 - Si **nameResolutionPhase** est **notStarted**, passer à l'étape 2).
 - Si **nameResolutionPhase** est **proceeding**, passer à l'étape 8).
 - Si **nameResolutionPhase** est **completed**, passer à l'étape 12).
- 2) Si une entrée a été trouvée (**lastEntryFound** de valeur différente de **0**), mettre la phase **nameResolutionPhase** à **proceeding** et passer à l'étape 9).
- 3) Si ce n'est pas le cas (**lastEntryFound = 0**), vérifier si le DSA est du premier niveau.

Si c'est le cas, la DSE racine ne contient pas de référence supérieure et dès lors elle n'est pas du type **supr.** Dans ce cas, passer à l'étape 4).

Si le DSA n'est pas du premier niveau, la DSE racine contient une référence supérieure et est donc du type **supr.** Dans ce cas, émettre une référence de continuation en utilisant la connaissance supérieure telle qu'elle est trouvée dans la DSE racine. Mettre:

 - **targetObject** au nom de l'objet cible;
 - **operationProgress.nameResolutionPhase** à **notStarted**;
 - **referenceType** à **superior**; et sélectionner
 - **accessPoints** selon les besoins.

Ajouter la référence de continuation à la liste des références de continuation dans **candidateRefs**. Passer à l'étape 6).
- 4) Vérifier si l'opération est dirigée vers l'entrée racine (**m = 0?**). Si c'est le cas, passer à l'étape 5). Si ce n'est pas le cas, émettre une référence de continuation en utilisant toute connaissance de NSSR trouvée dans la DSE racine. Mettre:
 - **targetObject** au nom de l'objet cible;
 - **operationProgress.nameResolutionPhase** à **proceeding**;
 - **operationProgress.nextRDNTToBeResolved** à **1**;
 - **referenceType** à **nonSpecificSubordinate**; et sélectionner
 - **accessPoints** selon les besoins.

Ajouter la référence de continuation à la liste des références de continuation de **candidateRefs**. passer à l'étape 6).
- 5) A une DSA de premier niveau, seule l'opération List ou Search peut être effectuée avec l'entrée racine comme objet de base. Pour cette raison et si l'opération n'était pas une List ou Search, retourner une **NameError** avec la cause **noSuchObject**. Si c'était une opération List ou Search, mettre **nameResolutionPhase** à **completed** et retourner le message **entry suitable**.
- 6) Vérifier s'il y a une quelconque référence de continuation dans **candidateRefs**. Si **candidateRefs** est vide, retourner une erreur **NameError** avec la cause **noSuchObject**. A défaut, passer à l'étape 7).
- 7) Utiliser une fonction de sélection locale pour choisir une référence de continuation dans la liste des références de continuation contenues dans **candidateRefs**, l'ajouter à la liste des références de continuation contenues dans **NRcontinuationList** et retourner le message **entry unsuitable**.
- 8) Si le DSA ne peut poursuivre la résolution du nom (auquel cas la valeur de **lastEntryFound** sera inférieure à celle de **nextRDNTToBeResolved**), passer à l'étape 11). Sinon passer à l'étape suivante.

- 9) Si la DSE(i) est une DSE dupliquée avec une connaissance subordonnée incomplète (**subordinateCompletenessFlag** à **FALSE**), émettre une référence de continuation à partir de l'attribut **supplierKnowledge** trouvé dans la **DSE (lastCP)**. Mettre:
- **targetObject** au nom de l'objet cible;
 - **operationProgress.nameResolutionPhase** à **proceeding**;
 - **operationProgress.nextRDNTToBeResolved** à **lastEntryFound**;
 - **referenceType** à **supplier**; et sélectionner
 - **accessPoints** selon les besoins.

Ajouter la référence de continuation à la liste des références de continuation contenues dans **NRcontinuationList** et retourner le message **entry unsuitable**.

- 10) Si la dernière entrée trouvée contient une NSSR (**DSE(lastEntryFound)** est de type **nssr**), émettre une référence de continuation à partir de la connaissance NSSR trouvée dans la **DSE(lastEntryFound)**. Mettre:
- **targetObject** au nom de l'objet cible;
 - **operationProgress.nameResolutionPhase** à **proceeding**;
 - **operationProgress.nextRDNTToBeResolved** à **lastEntryFound+1**;
 - **referenceType** à **nonSpecificSubordinate**; et sélectionner
 - **accessPoints** selon les besoins.

Ajouter la référence de continuation à la liste des références de continuation de **candidateRefs**. Passer à l'étape 7).

Si la **DSE(lastEntryFound)** n'est pas de type **nssr**, passer à l'étape 6).

- 11) Si **chainingArguments.referenceType** est de type **nssr**, passer à l'étape 13), sinon à l'étape 12).
- 12) Retourner une erreur **ServiceError** avec la cause **invalidReference**.
- 13) Si **i + 1** est égal à **nextRDNTToBeResolved**, la demande a été acheminée à cet endroit en raison d'une NSSR et le DSA n'a pas la capacité de poursuivre la résolution du nom; dans ce cas, retourner une **ServiceError** avec la cause **unableToProceed**; sinon, passer à l'étape 12).

18.3.3 Sous-procédure Target Found

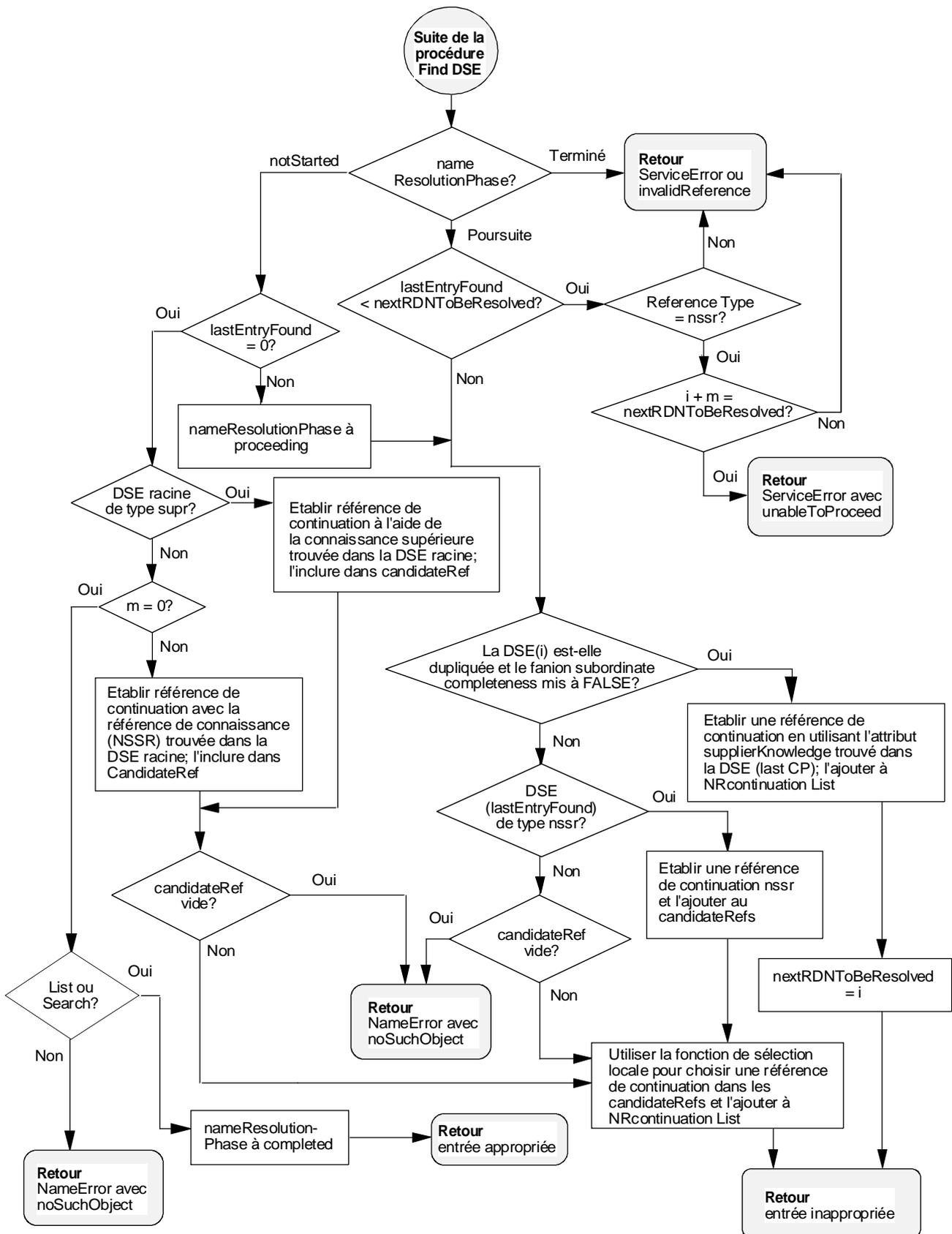
Cette sous-procédure est introduite quand le nom de l'objet cible correspond localement à une DSE; elle vérifie si l'entrée trouvée convient pour le traitement local de la demande et sa représentation est donnée à la Figure 11.

- 1) Appeler la procédure Check Suitability.
- 2) Si l'entrée est conforme (**entry suitable**), mettre **nameResolutionPhase** à **completed** et retourner le message **entry suitable**.
- 3) Si l'entrée n'est pas conforme (**entry unsuitable**), émettre une référence de continuation en utilisant l'attribut **supplierKnowledge** trouvé dans la **DSE(lastCP)**. Mettre:
 - **targetObject** au nom de l'objet cible;
 - **operationProgress.nameResolutionPhase** à **proceeding**;
 - **operationProgress.nextRDNTToBeResolved** à **i**;
 - **referenceType** à **supplier**; et sélectionner
 - **accessPoints** selon les besoins.

Ajouter la référence de continuation à la liste des références de continuation de **NRcontinuationList**. Retourner le message **entry unsuitable**.

NOTE – Si **serviceControl localScope** est à **TRUE**, le DSA peut, compte tenu des pratiques locales, décider de considérer cette entrée comme étant conforme et de procéder comme indiqué au point 2).

- 4) Si une extension critique n'est pas supportée (**unsupported critical extension**), retourner une erreur de type **ServiceError** avec la cause **unavailableCriticalExtension**.



TISO3690-94/d12

Figure 10 – Sous-procédure Target Not Found

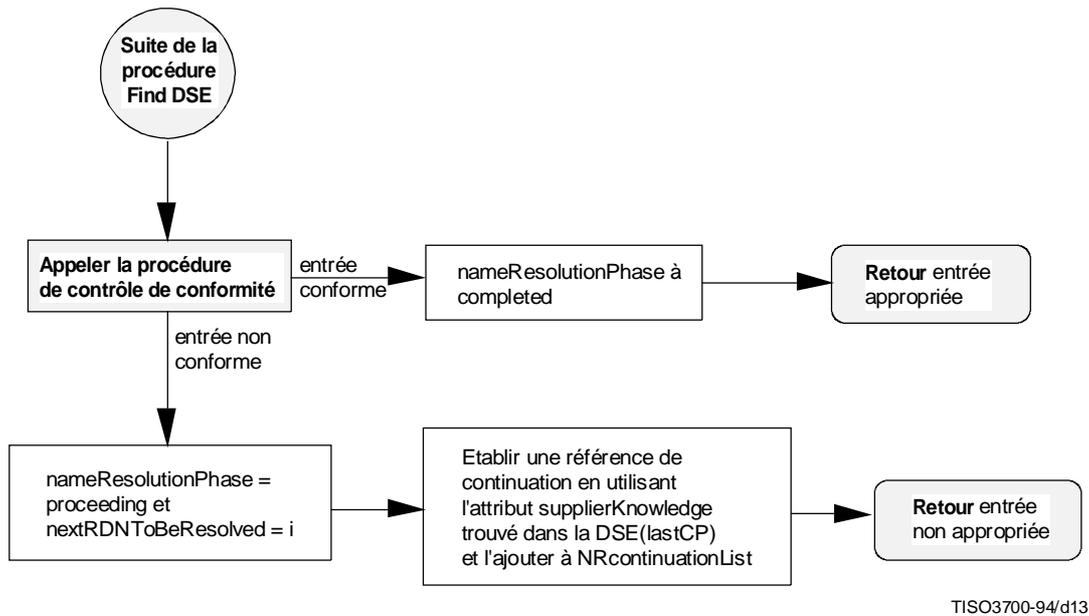


Figure 11 – Sous-procédure Target Found

18.3.4 Procédure Check suitability

Cette procédure est appelée pour décider si une DSE qui a été trouvée convient pour l'exécution de l'opération demandée. Elle tient compte des composants chainingArguments, serviceControls, des arguments fournis par l'utilisateur, du type d'opération et des caractéristiques de la DSE (duplication miroir, connaissance subordonnée, présence d'attributs, etc.).

18.3.4.1 Paramètres de la procédure

Les arguments d'entrée de cette procédure sont les suivants:

- une référence à une DSE;
- le type d'opération pour lequel la conformité de la DSE doit être vérifiée;
- le composant **chainingArguments**; et
- le composant serviceControls.

Le résultat est soit: entry suitable, entry unsuitable ou unsupported critical extension.

- 1) Si la DSE n'est pas du type **shadow**, vérifier si toutes les **criticalExtensions** (extensions critiques) sont supportées. Si c'est le cas, retourner entry suitable. Sinon, retourner unsupported critical extension.
- 2) Si la DSE est du type **shadow**, retourner entry unsuitable si l'une des conditions suivantes est vérifiée:
 - L'opération demandée est du type Modification.
 - La commande de service **dontUseCopy** est sélectionnée.
 Dans le cas contraire, passer à l'étape suivante.
- 3) Si la commande de service **copyShallDo** est sélectionnée, vérifier si toutes les **criticalExtensions** (extensions critiques) sont supportées. Si c'est le cas, retourner entry suitable. Sinon, retourner unsupported critical extension.
- 4) Si la commande de service **copyShallDo** n'est pas sélectionnée, vérifier si toutes les **criticalExtensions** (extensions critiques) sont supportées. Si c'est le cas, passer à l'étape 5); sinon, retourner entry unsuitable.
- 5) Faire la distinction entre les types d'opération suivants:
 - En cas d'opération List, passer à l'étape 6).

9594-4 : 1995 (F)

En cas d'opération Read, passer à l'étape 7).

S'il s'agit d'une opération Search ou Compare, passer à l'étape 8).

- 6) Si l'entrée bénéficie d'une connaissance subordonnée complète, l'opération List peut être effectuée. Dans ce cas, retourner entry suitable et, dans le cas contraire, entry unsuitable.
- 7) Si tous les attributs demandés sont présents dans la DSE, retourner entry suitable. Si certains attributs manquent, déterminer par des moyens locaux si la copie dupliquée détient tous les attributs détenus par le maître (par exemple en consultant l'accord de duplication miroir). Si c'est le cas, l'entrée est appropriée (retourner entry suitable). A défaut, le fournisseur peut détenir les attributs demandés qui ne sont pas présents dans la duplication miroir. Dans ce cas, la demande doit être chaînée (retourner entry unsuitable).
- 8) Si le DSA prend en compte la règle de concordance pour la comparaison ou la recherche, et que l'opération soit Compare ou Search avec **subset** de l'entrée **baseObject**, passer à l'étape 7). Si le DSA prend en compte la règle de concordance et qu'il s'agisse d'une opération Search avec le sous-ensemble **oneLevel** ou **subtree**, passer à l'étape 9). Dans le cas contraire, retourner entry unsuitable.
- 9) Si le composant composite **chainingArguments.excludeShadows** est à **True**, retourner entry unsuitable. Dans le cas contraire, vérifier la compréhension locale de la spécification de l'information dupliquée par rapport au filtre et au choix de l'opération. En présence de toutes les entrées et de tous les attributs nécessaires, retourner entry suitable. Si une entrée ou un attribut manque, retourner un message entry unsuitable.

19 Evaluation de l'opération

Le présent article définit la procédure que doit suivre un DSA si l'entrée cible d'une opération a été trouvée sur place (au cours de la résolution du nom). Suivant le type d'opération, l'une des procédures suivantes est invoquée:

- pour une opération **AddEntry**, **ChainedAddEntry**, **RemoveEntry**, **ChainedRemoveEntry**, **ModifyEntry**, **ChainedModifyEntry**, **ModifyDN** ou **ChainedModifyDN**, suivre les procédures du 19.1;
- pour une opération **Read**, **ChainedRead**, **Compare** ou **ChainedCompare**, suivre les procédures du 19.2;
- pour une opération **Search**, **ChainedSearch**, **List** et **ChainedList**, suivre les procédures du 19.3.

19.1 Procédure de modification

Selon le type d'opération de modification, suivre les procédures correspondantes définies aux 19.1.1 à 19.1.4.

19.1.1 Opération AddEntry

- 1) Le DSA s'assure que l'initiateur a suffisamment de droits d'accès, comme défini au 11.1.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. Si ce n'est pas le cas, l'erreur appropriée est retournée.
- 2) Le DSA s'assure qu'une entrée ayant le nom de l'entrée à ajouter n'existe pas encore; si elle existe, il doit retourner une erreur **UpdateError** avec la cause **entryAlreadyExists**. Si la DSE est de type additionnel de **nssr**, le DSA doit suivre la procédure définie au 19.1.5 (Opérations de modification et NSSR) pour s'assurer que le nom de la nouvelle entrée n'est pas ambigu.
- 3) Si le composant **targetSystem** est présent et que le point **AccessPoint** ne soit pas celui du DSA actuel, passer à l'étape 4). Si le composant **targetSystem** n'est pas présent ou s'il est présent et que **AccessPoint** soit celui du DSA actuel, passer à l'étape 5).
- 4) Si l'entrée est une sous-entrée, le DSA doit retourner l'erreur **UpdateError** avec la cause **affectsMultipleDSAs**. Si l'entrée n'est pas une sous-entrée, le DSA peut choisir localement si oui ou non il souhaite établir un HOB avec le DSA spécifié. Si ce n'est pas le cas, le DSA doit retourner une erreur de type **ServiceError** avec la cause **unwillingToPerform**; sinon il doit établir un rattachement opérationnel hiérarchique avec le DSA subordonné spécifié. Si le DOP est supporté, la procédure du 24.3.1.1 doit être suivie; sinon, des moyens locaux sont utilisés pour établir le HOB. Si le DSA subordonné ne souhaite pas l'établissement du rattachement opérationnel, une erreur de type **ServiceError** avec la cause **unwillingToPerform** est retournée pour l'opération **addEntry**. Si le HOB est normalement établi, passer à l'étape 7).

NOTE 1 – Cette étape de la procédure n'est pas applicable à la création de zones administratives autonomes chez un DSA subordonné.

- 5) Le DSA doit veiller à ce que la nouvelle entrée ou sous-entrée à ajouter soit conforme au sous-schéma ou au schéma du système, selon le cas (par exemple que la DSE immédiatement supérieure soit du type **admPoint**). Si elle n'est pas conforme, il doit retourner une erreur de type approprié (**UpdateError** ou **AttributeError**) ou bien ajouter la nouvelle entrée/sous-entrée. S'il s'agit d'une entrée, passer à l'étape 7); s'il s'agit d'une sous-entrée, passer à l'étape 6).
- 6) Le DSA doit expédier, à un moment approprié, un rattachement opérationnel de modification à tous les DSA subordonnés avec lesquels il possède des rattachements opérationnels hiérarchiques ou hiérarchiques non spécifiques applicables. Les rattachements applicables sont ceux qui sont associés à des contextes de dénomination subordonnés à la DSE supérieure. Les contextes de dénomination dont les préfixes correspondent à des points administratifs autonomes ne sont pas applicables. Si le DOP est supporté, les procédures indiquées aux 24.3.2.1 et 25.3.2 doivent être suivies. Si le DOP n'est pas supporté, des moyens locaux doivent être utilisés pour modifier les RHOB.

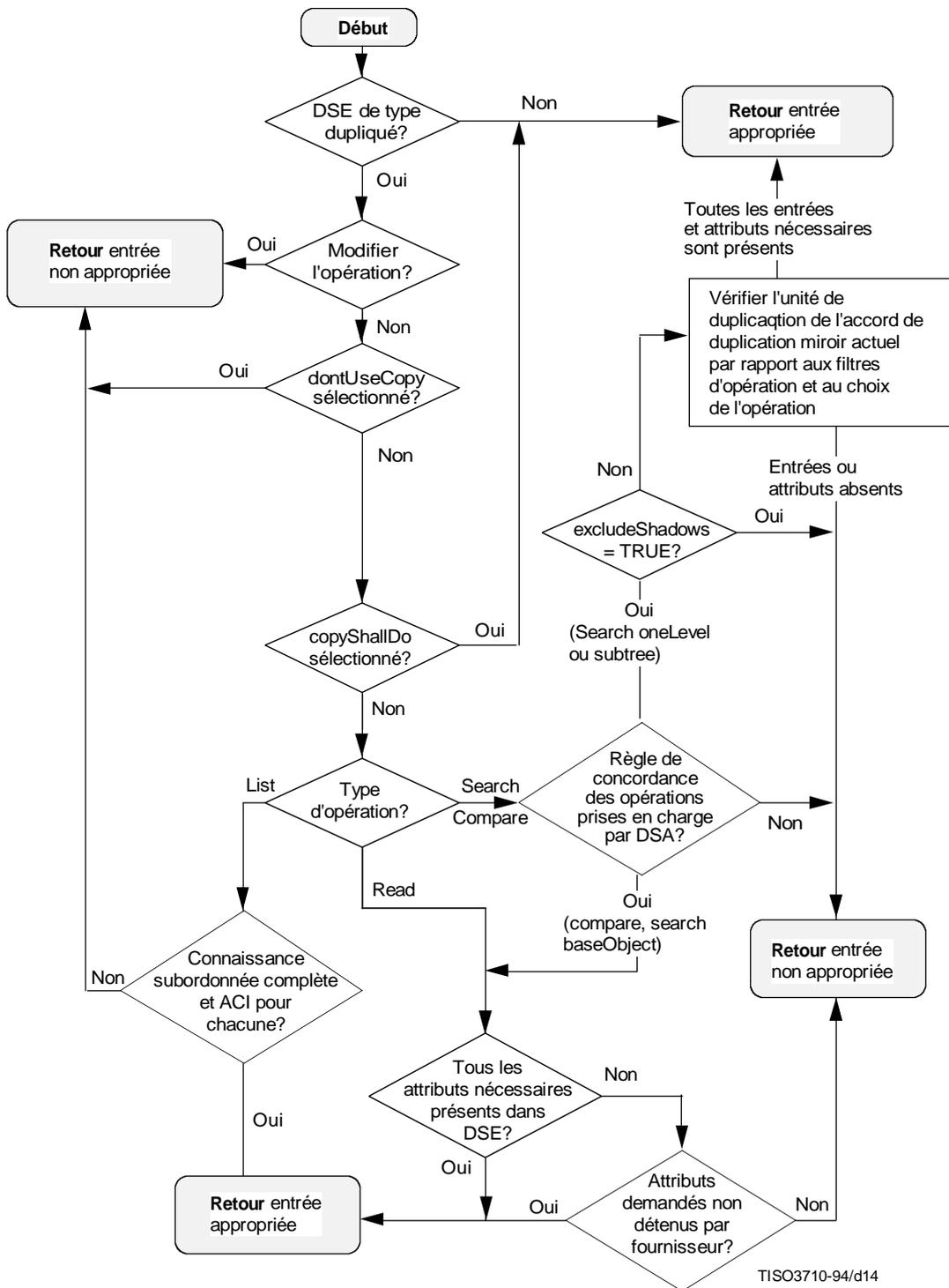
NOTE 2 – Un moment approprié est spécifié par l'administrateur des DSA. Il peut s'insérer immédiatement après (ou même un peu avant) le retour du résultat d'opération selon une stratégie de récurrence (par exemple à une heure de consigne). Le moment peut varier selon le motif de la modification, par exemple des mises à jour d'informations de contrôle d'accès (ACI) prenant effet immédiatement et des modifications périodiques du schéma.

- 7) Si l'entrée ou sous-entrée ajoutée fait partie du composant **UnitOfReplication** d'un ou de plusieurs accords de duplication miroir, les consommateurs d'informations miroirs doivent être mis à jour au moyen des procédures du service de duplication miroir d'Annuaire spécifié dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.

19.1.2 Opération Remove Entry (Suppression d'entrée)

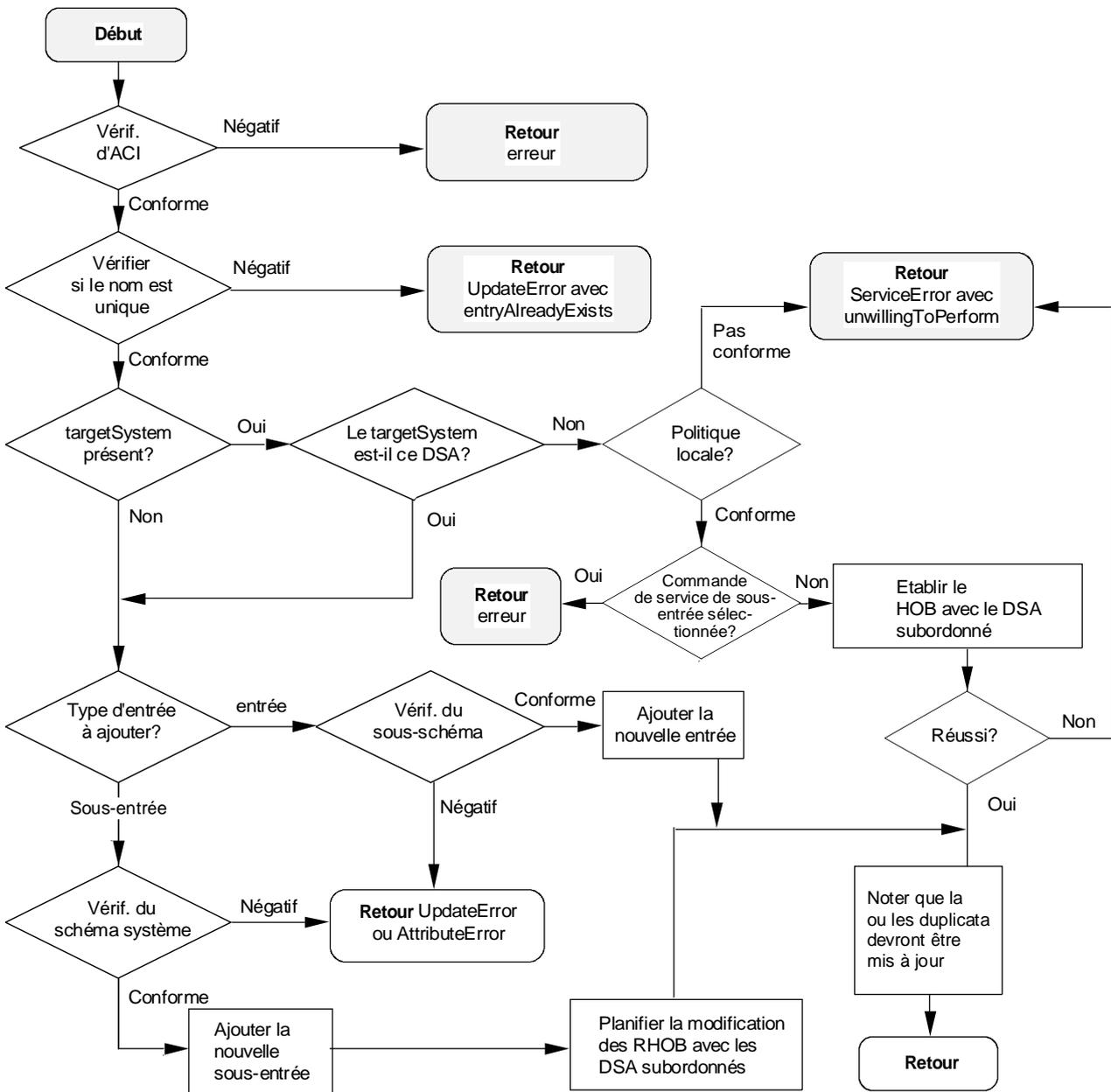
- 1) Le DSA doit s'assurer que l'initiateur a suffisamment de droits d'accès, par exemple ceux qui sont définis au 11.2.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. Si ce n'est pas le cas, une erreur appropriée est retournée.
- 2) Le DSA doit s'assurer que l'entrée à supprimer est une entrée feuille. Si ce n'est pas le cas, le DSA retourne une **UpdateError** avec la cause **notAllowedOnNonLeaf**.
- 3) Le type de DSE est contrôlé sur l'entrée à supprimer. S'il s'agit du type **subentry**, passer à l'étape 5). S'il s'agit du type **cp**, passer à l'étape 6). S'il s'agit du type **entry** ou **alias**, passer à l'étape 4).
- 4) Supprimer l'entrée ou l'alias et passer à l'étape 7).
- 5) Supprimer la sous-entrée. A un moment approprié, modifier les rattachements opérationnels hiérarchiques de tous les DSA subordonnés pertinents avec lesquels le DSA actuel possède des rattachements opérationnels hiérarchiques ou hiérarchiques non spécifiques. Les rattachements pertinents sont ceux qui sont associés aux contextes de dénomination subordonnés à la DSE supérieure. Les contextes de dénomination dont les préfixes correspondent à des points administratifs autonomes ne sont pas pertinents. Si le DOP est pris en charge, les procédures des 24.3.2.1 et 25.3.2 doivent être suivies. A défaut, des moyens locaux doivent être utilisés. Passer à l'étape 7).
- 6) Supprimer le contexte de dénomination. Si le DSA a un rattachement opérationnel hiérarchique pour ce contexte de dénomination, il met fin à ce rattachement avec son DSA immédiatement supérieur. Si le DSA a un rattachement opérationnel hiérarchique non spécifique pour ce contexte de dénomination et qu'il s'agisse du dernier contexte de dénomination du rattachement en question, il met fin à ce rattachement avec son DSA immédiatement supérieur. Si le DOP est pris en charge, les procédures des 24.3.3.2 et 25.3.3.2 doivent être suivies. Sinon, des moyens locaux sont utilisés pour mettre fin au RHOB.
- 7) Si la suppression du contexte de dénomination, de l'entrée, de l'entrée alias ou de la sous-entrée faisait partie du composant **UnitOfReplication** d'un ou de plusieurs accords de duplication miroir, la liste des consommateurs d'informations miroirs doit être mise à jour au moyen des procédures du service de duplication miroir d'Annuaire spécifié dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Si la suppression de la référence subordonnée ou subordonnée non spécifique détenue par le DSA immédiatement supérieur (dont le RHOB a été arrêté) faisait partie du composant **UnitOfReplication** d'un ou de plusieurs accords de duplication miroir, la liste des consommateurs d'informations miroirs doit être mise à jour au moyen des procédures du service de duplication miroir d'Annuaire spécifié dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.



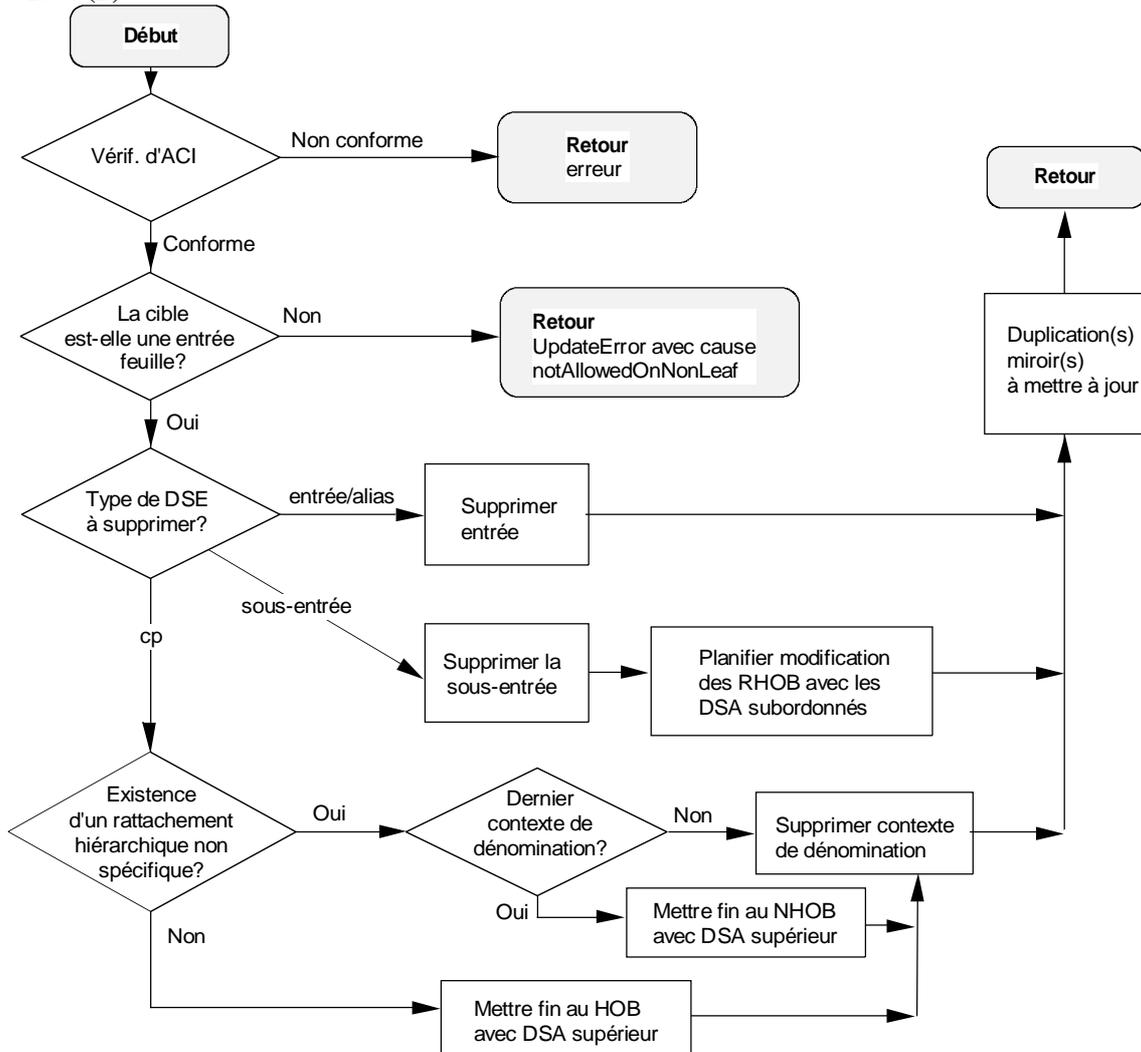
TISO3710-94/d14

Figure 12 – Procédure Check Suitability



TISO3720-94/d15

Figure 13 – Procédure Add Entry



TISO3730-94/d16

Figure 14 – Procédure Remove Entry

19.1.3 Opération Modify Entry (Modification d'entrée)

- 1) Le DSA doit s'assurer que l'initiateur dispose de droits d'accès comme ceux qui ont été définis au 11.3.5. de la Rec. UIT-T X.511 | ISO/CEI 9594-3. Si ce n'est pas le cas, une erreur appropriée est retournée.
- 2) Les modifications à l'entrée, à l'entrée alias ou à la sous-entrée seront conformes respectivement au schéma ou au sous-schéma du système; à défaut, le DSA doit retourner l'erreur appropriée, **UpdateError** ou **AttributeError**. Après avoir exécuté les modifications, si la DSE cible est de type **subentry**, passer à l'étape 3); si elle est d'un autre type, passer à l'étape 4).
- 3) Le DSA doit, à un moment approprié, modifier les rattachements opérationnels à tous les DSA subordonnés pertinents avec lesquels il entretient des rattachements opérationnels hiérarchiques ou hiérarchiques non spécifiques. Les rattachements pertinents sont ceux qui sont établis avec des contextes de dénomination subordonnés au point administratif dont dépend la sous-entrée modifiée. Les contextes de dénomination dont les préfixes correspondent à des points administratifs autonomes ne sont pas pertinents. Si le DOP est pris en charge, suivre la procédure du 24.3.2.1 et du 25.3.2. Sinon, utiliser des moyens locaux.

- 4) Si la modification d'entrée, d'entrée alias ou de sous-entrée faisait partie du composant **UnitOfReplication** d'un ou de plusieurs accords de duplication miroir, la liste des consommateurs d'informations miroirs doit être mise à jour au moyen des procédures du service de duplication miroir d'informations d'Annuaire spécifié dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.

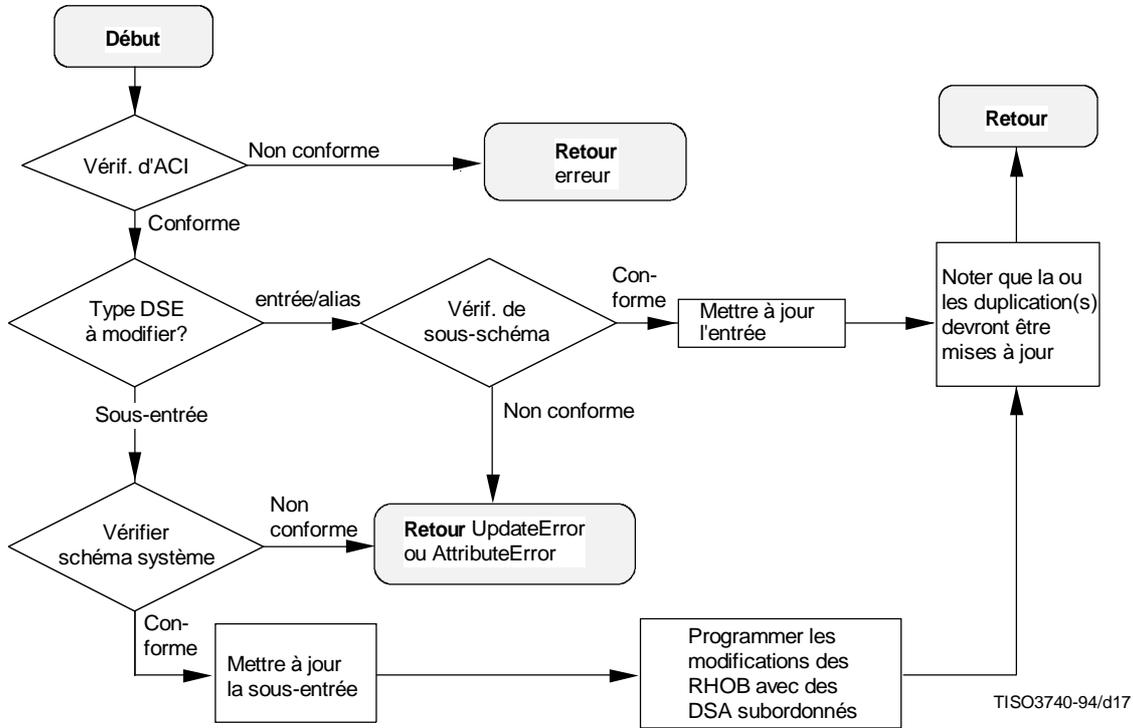


Figure 15 – Procédure Modify Entry

19.1.4 Opération Modify DN (Modification de nom distinctif)

- 1) Le DSA doit s'assurer que l'initiateur dispose de suffisamment de droits d'accès comme ceux qui sont définis au 11.4.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. Si ce n'est pas le cas, l'erreur appropriée est retournée.
- 2) Si l'opération consiste à déplacer une entrée vers un nouveau supérieur au sein du même DSA, passer à l'étape 3). Si l'opération consiste à modifier le nom distinctif relatif (RDN) d'une entrée, passer à l'étape 4).
- 3) L'opération doit être effectuée conformément à la définition du 11.4.1 de la Rec. UIT-T X. 511 | ISO/CEI 9594-3. Si soit l'ancien supérieur, le nouveau supérieur, l'entrée ou un de ses subordonnés détient des NSSR, l'opération doit être rejetée avec un message d'erreur de type **UpdateError** et de cause **AffectsMultipleDSAs**. Sinon, déplacer les entrées dans le DSA et passer à l'étape 9).
- 4) Le texte qui suit s'applique aux changements du nom distinctif relatif d'entrée/sous-entrée/alias, qui peut ou ne peut pas être une entrée feuille et qui peut ou non avoir une ou plusieurs subordonnées dans un ou plusieurs DSA. Le type de DSE est vérifié pour l'entrée à renommer. S'il s'agit du type **subentry**, passer à l'étape 7). S'il s'agit du type **cp**, passer à l'étape 6). S'il s'agit du type **entry** ou **alias**, passer à l'étape 5).

9594-4 : 1995 (F)

- 5) Le DSA doit s'assurer qu'il n'existe pas d'autres entrées portant le nouveau nom; si une telle entrée existe, il doit retourner une erreur de type **UpdateError** avec la cause **entryAlreadyExists**. Si la DSE supérieure de l'entrée à renommer est de type additionnel **nssr**, le DSA doit suivre la procédure définie au 19.1.5 (Opérations de modification et NSSR) pour s'assurer que le nouveau nom de l'entrée n'est pas ambigu. Le DSA doit s'assurer que le nouveau nom de l'entrée ou de la sous-entrée est conforme au schéma ou sous-schéma du système (selon le cas), à défaut de quoi il doit retourner une erreur **AttributeError** ou **UpdateError**, selon le cas. Renommer l'entrée ou l'entrée alias. Si l'entrée n'est pas une entrée feuille et possède des subordonnées dans d'autres DSA, passer à l'étape 8). Sinon, passer à l'étape 9).
- 6) Le DSA doit s'assurer que le nouveau nom du contexte de dénomination est conforme au sous-schéma. Sinon, il doit retourner une erreur appropriée, de type **AttributeError** ou **UpdateError**.

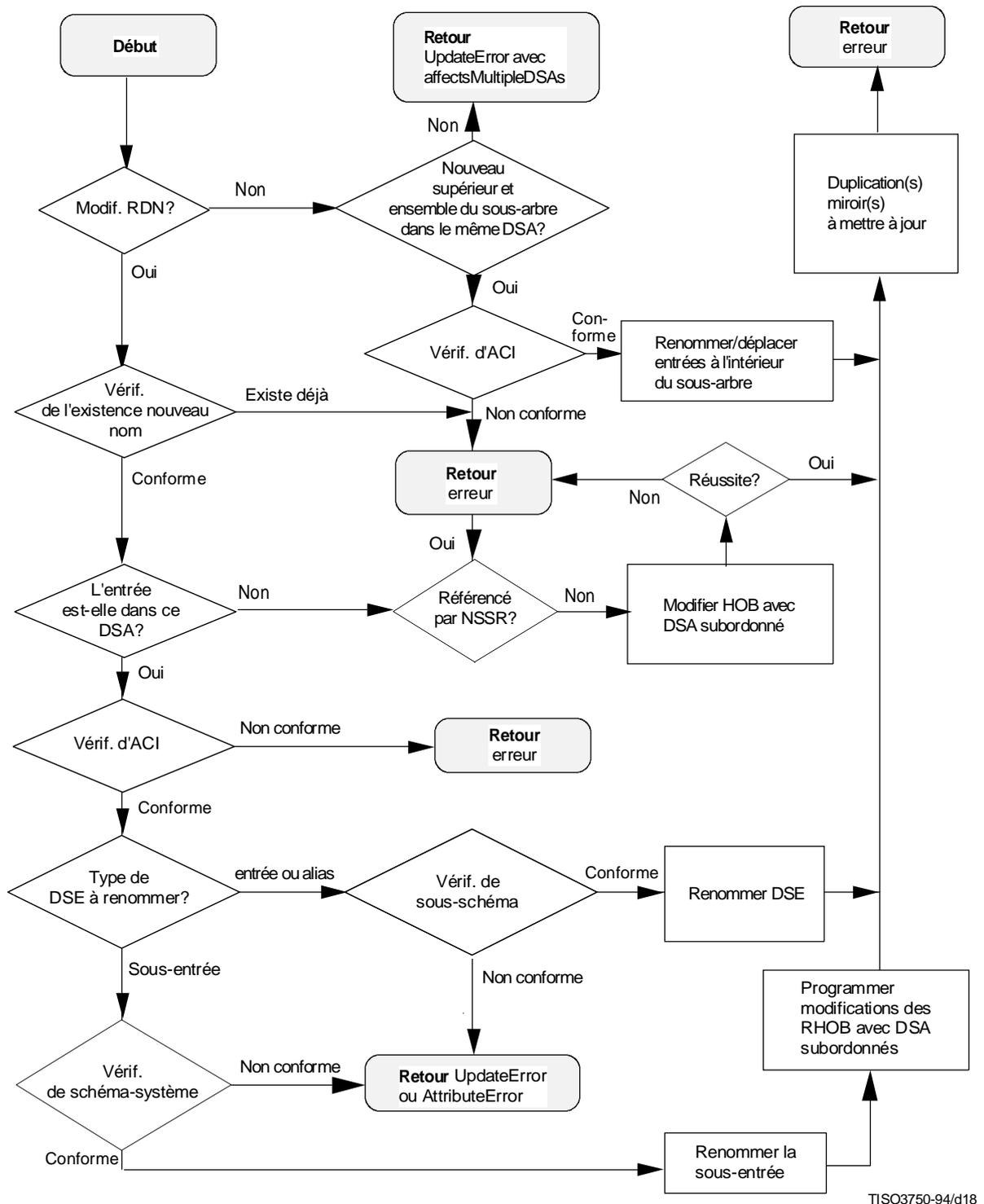
Si le DSA possède un HOB avec le DSA supérieur, ce DSA subordonné doit tenter de modifier ce HOB avant de répondre à l'opération de modification du nom distinctif. Le DSA supérieur doit veiller, avant d'accepter la modification, à ce qu'aucune autre entrée portant le nouveau nom n'existe déjà. Si le DOP est pris en charge, la procédure du 24.3.2.2 doit être suivie. S'il n'est pas pris en charge, la façon de modifier le HOB et la vérification de l'unicité du nouveau nom relèveront de moyens locaux. Si le HOB est bien modifié et que le contexte de dénomination possède des contextes subordonnés dans d'autres DSA, passer à l'étape 8). Sinon, passer à l'étape 9). Si le HOB ne peut pas être modifié, retourner une erreur du type **UpdateError** avec la cause **affectsMultipleDSAs**.

Si le DSA possède un NHOB avec le DSA supérieur pour ce contexte de dénomination, la façon dont les entrées dupliquées sont détectées n'est pas dans le domaine d'application de la présente Spécification d'Annuaire. Renommer l'entrée. Si le contexte de dénomination a des contextes subordonnés dans d'autres DSA, passer à l'étape 8). Sinon, passer à l'étape 9).

- 7) Le DSA doit s'assurer que le nouveau nom de la sous-entrée est conforme au schéma du système. Sinon, il doit retourner une erreur appropriée: **AttributeError** ou **UpdateError**. Le DSA doit s'assurer qu'aucune autre sous-entrée portant le nouveau nom n'existe déjà. Sinon, il doit retourner une erreur de type **UpdateError** avec la cause **entryAlreadyExists**.
- 8) Le DSA doit, à un moment approprié, modifier les rattachements opérationnels à tous les DSA subordonnés avec lesquels il a des rattachements opérationnels hiérarchiques ou hiérarchiques non spécifiques. Les rattachements pertinents sont ceux qui sont associés à tous les contextes de dénomination subordonnés à la sous-entrée modifiée ou aux contextes de dénomination pertinents qui sont subordonnés au point administratif dont la sous-entrée a été renommée. Les contextes de dénomination dont les préfixes correspondent à des points administratifs autonomes ne sont pas pertinents. Si le DOP est pris en charge, suivre la procédure du 24.3.2.1 et du 25.3.2. Sinon, utiliser des moyens locaux pour mettre à jour les RHOB.
- 9) Si le renommage du contexte de dénomination, de l'entrée ou d'une de ses subordonnées, de l'entrée alias ou de la sous-entrée fait partie du composant **UnitOfReplication** d'un ou de plusieurs accords de duplication miroir détenus par le DSA, la liste des consommateurs d'informations miroirs doit être mise à jour au moyen des procédures du service de duplication miroir d'informations d'Annuaire spécifié dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Si la référence subordonnée renommée dans le DSA immédiatement supérieur [dont le HOB a été modifié à l'étape 6) ci-dessus], fait partie du composant **UnitOfReplication** d'un ou de plusieurs accords de duplication miroir détenus par le DSA, la liste des consommateurs d'informations miroirs doit être mise à jour au moyen des procédures du service de duplication miroir d'informations d'Annuaire spécifié dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Si les composants d'un RHOB à un DSA subordonné [tel que modifié à l'étape 8) ci-dessus] font partie du composant **UnitOfReplication** d'un ou de plusieurs accords de duplication miroir détenus par le DSA, la liste des consommateurs d'informations miroirs doit être mise à jour au moyen des procédures du service de duplication miroir d'informations d'Annuaire spécifié dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.



TISO3750-94/d18

Figure 16 – Procédure Modify DN

9594-4 : 1995 (F)

19.1.5 Opérations de modification et références subordonnées non spécifiques (NSSR)

Si un DSA a des NSSR mais qu'il ne connaisse pas l'ensemble complet des noms des subordonnés d'une entrée vers laquelle:

- a) soit une opération **AddEntry** a été envoyée;
- b) soit une opération **ModifyDN** a été envoyée;

le DSA en question peut exécuter la série suivante de procédures avant d'exécuter l'opération.

- 1) Si la commande de service **chainingProhibited** est mise à l'opération **AddEntry** ou **ModifyDN**, le DSA retourne une erreur de type **UpdateError** avec la cause **affectsMultipleDSAs**.
- 2) Si le DSA n'est ni disposé ni en mesure de multichaîner les demandes sortantes, il doit retourner une erreur de type **ServiceError** avec, selon le cas, la cause **unwillingToPerform** ou **unavailable**.
- 3) Le DSA doit multichaîner une opération **ChainedReadEntry** vers chacun des DSA de l'ensemble **accessPointInformation** de la NSSR (le DSA utilise uniquement le DSA maître de chaque variable **MasterAndShadowAccessPoints** en raison de l'incohérence passagère résultant de la duplication miroir). Les paramètres de la variable **ReadArgument** seront réglés comme suit:

object mis soit au nom de l'entrée à ajouter (dans le cas de l'opération **AddEntry**) soit au nom proposé d'une entrée existante (dans le cas de l'opération **ModifyDN**)

selection l'attribut de la classe d'objets

Les paramètres de la variable **CommonArguments** seront réglés comme suit:

- **ServiceControls.options** mis à **dontDereferenceAliases**;
- **OperationProgress.nameResolutionPhase** mis à **completed**.

Les paramètres de la variable **ChainingArguments** seront réglés comme suit:

- **originator** mis au nom de l'expéditeur;
- **targetObject** omis;
- **OperationProgress.nameResolutionPhase** mis à **proceeding** et **nextRDNTToBeResolved** mis à (nombre de RDN contenus dans le nom objet)–1;
- **traceInformation** mis à une séquence vide;
- **referenceType** mis à **nonSpecificSubordinate**.

Les autres paramètres, comme **SecurityParameters**, pourront être réglés selon les besoins, par exemple selon une politique locale.

- 4) Le DSA attend la série complète de réponses. Si l'une d'elles est de type **ReadResult**, une erreur est retournée comme à l'étape 6) ci-dessous.
- 5) Si toutes les réponses sont du type **ServiceError** avec la cause **unableToProceed**, l'évaluation de l'opération peut se poursuivre.
- 6) En cas de retour d'un résultat **ReadResult**, une erreur **UpdateError** avec la cause **entryAlreadyExists** sera retournée pour l'opération initiale.
- 7) Si une quelconque autre erreur est retournée à la demande de **ReadEntry**, une erreur **ServiceError** avec la cause **unwillingToPerform** est retournée.

Le DSA recevant une demande **ChainedReadRequest** donnera une réponse selon la présence ou non de l'entrée et compte tenu de sa politique de contrôle d'accès.

19.2 Procédure d'interrogation à entrée unique

Les opérations **Read**, **ChainedRead**, **Compare** et **ChainedCompare** relèvent du groupe de procédures d'interrogation à entrée unique. Ces procédures contiennent uniquement les trois étapes ci-dessous:

- 1) vérification de contrôle d'accès, comme décrit à l'article 10 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. Si l'opération est désapprouvée, retourner l'erreur de sécurité appropriée;
- 2) effectuer l'opération sur la DSE trouvée comme indiquée à l'article 9 de la Rec. UIT-T X.511 | ISO/CEI 9594-3;
- 3) préparer la réponse et la retourner.

19.3 Procédure d'interrogation à entrées multiples

Selon le type d'opération d'interrogation (**List** ou **Search**), les procédures correspondantes, définies aux 19.3.1 et 19.3.2, s'appliquent.

19.3.1 Procédure List

Le présent paragraphe contient la spécification de la procédure d'évaluation relative aux opérations **List** et **ChainedList**.

La procédure **List Procedure (I)** doit être suivie quand le composant **operationProgress.name-ResolutionPhase** de la demande est mis à **notStarted** ou à **proceeding** ou quand le DSA, après avoir effectué la résolution du nom, constate qu'il contient l'entrée d'objet de base. La procédure **List Procedure (II)** doit être suivie quand le composant **nameResolutionPhase** de la demande de listage a été mis à **completed**.

19.3.1.1 Paramètres de la procédure

19.3.1.1.1 Arguments

Les arguments utilisés par cette procédure sont les suivants:

- **Argument**;
- l'entrée cible DSE **e**;
- le paramètre **operationProgress** du composant **chainingArgument**.

19.3.1.1.2 Résultats

Si la procédure aboutit, elle retourne:

- un ensemble de subordonnés de **e** dans **listInfo.subordinates**;
- la cause **limitProblem** indiquée dans **partialOutcomeQualifier**;
- un ensemble de références de continuation dans **SRcontinuationList**.

19.3.1.1.3 Erreurs

La procédure peut aboutir à l'une des erreurs suivantes, retournées au DUA/DSA demandeur:

- une **accessControlError** ou **nameProblem**;
- toute erreur définie pour la procédure **Find DSE** quand un alias a été déréférencé.

19.3.1.2 Définition de la procédure

Les sous-procédures définies aux 19.3.2.2.1 et 19.3.2.2.2 sont invoquées en fonction des règles suivantes:

19.3.1.2.1 List Procedure (I)

La List Procedure (I) est constituée des étapes suivantes, illustrées à la Figure 17.

- 1) Si la commande de service **subentry** est sélectionnée, ajouter au composant composite **listResult.subordinates**, pour chaque sous-entrée dont l'accès est autorisé, les RDN des DSE de type **subentry** immédiatement subordonnées à l'entrée **e**. Si aucun accès à une quelconque entrée n'est permis, ignorer cette entrée.
- 2) Si la DSE **e** est de type **nssr**, ajouter une **continuationReference** à **SRcontinuationList** avec les composants suivants:
 - **targetObject** mis au nom de la DSE **e**;
 - **aliasedRDNs** absent;
 - **operationProgress** avec **nameResolutionPhase** mis à **completed** et **nextRDNTobeResolved** absent;
 - **rdnsResolved** absent;
 - **referenceType** mis à **nonSpecificSubordinate**;
 - **accessPoints** mis à un ensemble d'items **accessPointInformation** tirés chacun d'une valeur de l'attribut **nonSpecificKnowledge** de la DSE **e**.

9594-4 : 1995 (F)

- 3) Pour chaque entrée DSE **e'** immédiatement subordonnée à la DSE **e**, exécuter les opérations suivantes:
- a) Vérifier dans la sous-entrée **e'** si l'information ACI est disponible. Si l'information ACI interdit le listage des RDN de **e'**, omettre cette DSE. Si l'information ACI n'est pas disponible (par exemple en cas de référence subordonnée et d'entrée interstitielle), poursuivre ou non compte tenu de la politique locale.
 - b) Vérifier tous les types de DSE de **e'**
 - i) Si **e'** est du type **subr**, deux cas sont possibles. Dans le premier, l'information ACI de l'entrée subordonnée et la classe d'objet sont disponibles sur place; dans ce cas, conformément à la politique locale et aux informations ACI, ajouter le RDN de la sous-entrée **e'** à **listResult.subordinates** avec **aliasEntry** mis à **True** si **e'** est de type **sa** et **fromEntry** mis à **False**. L'autre cas est celui où l'entrée de l'information ACI n'est pas disponible dans **e'**; dans ce cas ajouter une référence **continuationReference** à la liste **SRcontinuationList** avec les composants suivants:
 - **targetObject** mis au nom de la DSE **e**;
 - **aliasedRDN** absents;
 - **operationProgress** avec **nameResolutionPhase** mis à **completed** et **nextRDNTobeResolved** absent;
 - **rdnsResolved** absent;
 - **referenceType** mis à **subordinate**;
 - **accessPoints** mis à la valeur contenue dans l'attribut **specificKnowledge** de la sous-entrée DSE **e'**.
 - ii) Si la sous-entrée DSE **e'** est de type **entry** ou **glue** (interstitielle), ajouter le RDN de **e'** à **listResult.subordinates**, avec **aliasEntry** mis à **False** et **fromEntry** sélectionné selon que **e'** est ou non une copie.

NOTE – Si l'entrée **e'** est interstitielle (**glue**), elle doit avoir un ou plusieurs subordonnés, ce qui implique qu'elle ne peut être un alias dans le DSA maître. Par ailleurs, toute information ACI applicable au listage est enregistrée dans cette DSE, fournie via le protocole de duplication miroir.
 - iii) Si la sous-entrée DSE **e'** est de type **alias**, ajouter le RDN de l'entrée **e'** à **listResult.subordinates** avec **aliasEntry** mis à **True**, et **fromEntry** sélectionné si **e'** est une copie.
 - c) Vérifier si l'heure, la taille ou la limite administrative est dépassée. Si c'est le cas, sélectionner **limitProblem** en conséquence dans **partialOutcomeQualifier** et faire le retour.
 - d) Poursuivre à partir du point 3), a) jusqu'à ce que toutes les DSE subordonnées aient été traitées.
- 4) Quand toutes les DSE subordonnées ont été traitées, retourner à l'aiguilleur d'opérations.

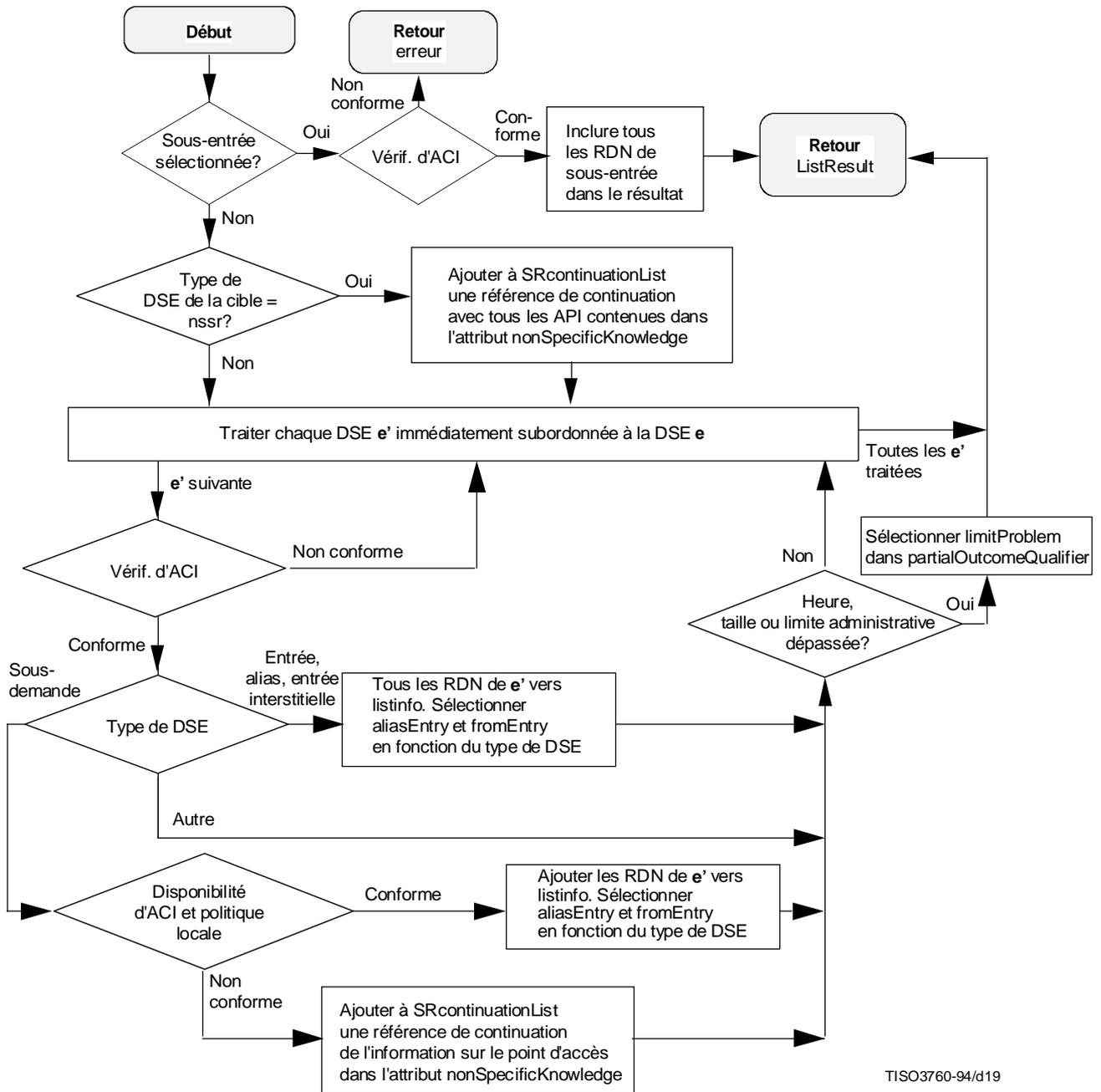
19.3.1.2.2 Procédure List Procedure (II)

La procédure List Procedure (II) est constituée des étapes suivantes, illustrées à la Figure 18:

- 1) Pour chaque sous-entrée DSE **e'** immédiatement subordonnée à la DSE **e**, exécuter les opérations 1), a) à 1), d) suivantes:
 - a) Vérifier l'information ACI dans **e'**. Si l'opération est interdite par l'information ACI, poursuivre avec la subordonnée immédiatement suivante de **e**;
 - b) ajouter le RDN de la sous-entrée DSE **e'** au composant composite **listResult.subordinates**, le composant **aliasEntry** de **listResult.subordinates** étant sélectionné si **e'** est un alias et le composant **fromEntry** étant sélectionné selon que **e'** est ou non une copie. Si le composant **excludeShadows** a la valeur **TRUE**, ne pas tenir compte des DSE de type **shadow**.
 - c) vérifier si l'heure, la taille ou la limite administrative est dépassée. Si oui, sélectionner **limitProblem** de **partialOutcomeQualifier** en conséquence et retourner le résultat;
 - d) poursuivre l'étape 1), a) jusqu'à ce que toutes les DSE subordonnées aient été traitées.

- 2) Quand toutes les DSE subordonnées ont été traitées, vérifier si cette sous-demande a été acheminée par DAP ou DSP. Si cette sous-demande est soumise via le DAP et que **listResult** soit vide, retourner une **serviceError** avec la cause **invalidReference** à l'ailleur d'opérations. Sinon, retourner **listResult**.

NOTE – La référence **invalidReference** s'utilise à titre de sécurité si l'utilisateur n'a pas accès à l'entrée supérieure. Si l'information de contrôle d'accès (ACI) de l'entrée supérieure est disponible (fournie par le rattachement opérationnel pertinent RMOB), un résultat nul peut être retourné si cela est permis.



TISO3760-94/d19

Figure 17 – Procédure List (I)

9594-4 : 1995 (F)

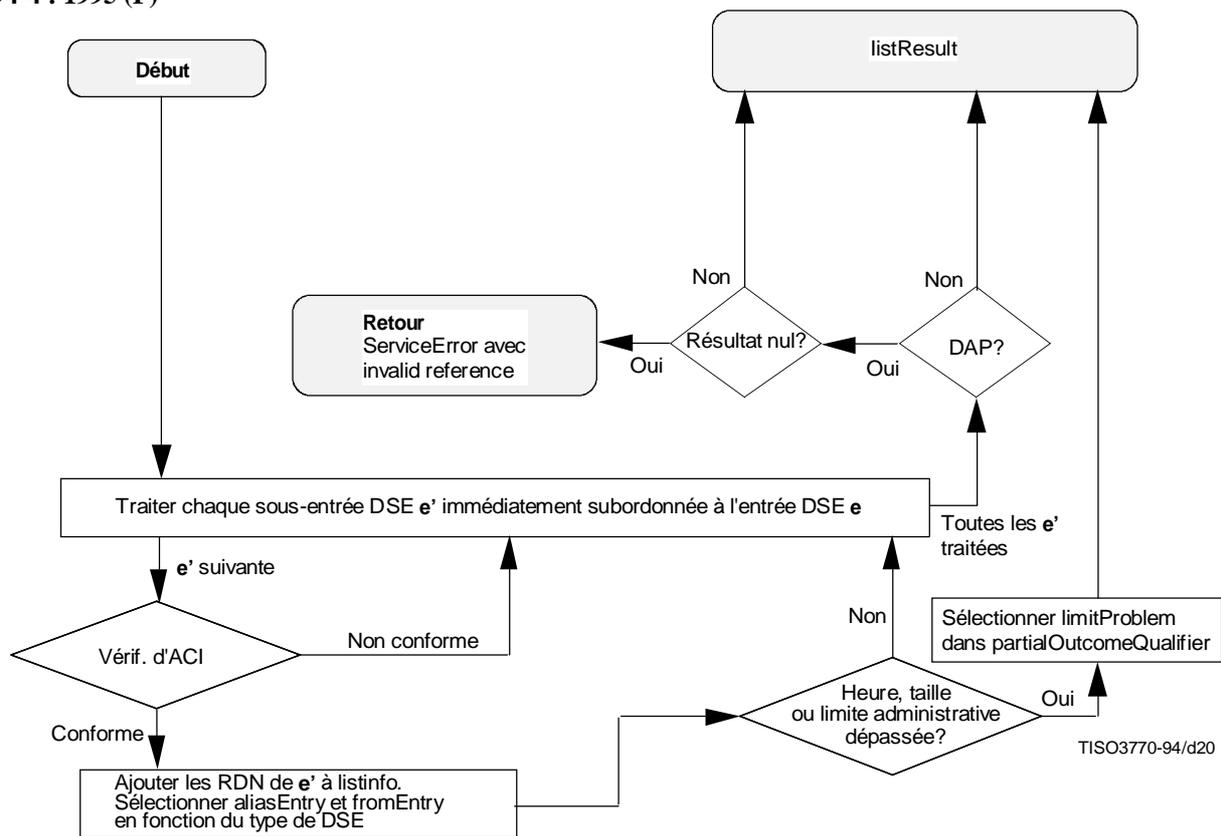


Figure 18 – Procédure List (II)

19.3.2 Procédure Search

Le présent paragraphe spécifie la procédure d'évaluation spécifique aux opérations **Search** et **ChainedSearch**.

La procédure **Search** (I) doit être suivie quand le composant composite **operationProgress.nameResolutionPhase** de demande de recherche n'est pas mis à **notStarted** ou à **proceeding** et quand le DSA, après avoir effectué la résolution du nom, estime qu'il détient l'entrée d'objet de base. La procédure **Search Procedure (II)** doit être suivie quand le composant **nameResolutionPhase** de la demande de recherche est mis à **completed**.

NOTE – Quand la variable **nameResolutionPhase** est à **completed**, on présume que l'objet cible est le supérieur immédiat d'un préfixe de contexte.

19.3.2.1 Paramètres de la procédure

19.3.2.1.1 Arguments

Les arguments qui sont utilisés pour cette procédure sont les suivants:

- **searchArgument**;
- la DSE **e** cible;
- le paramètre **operationProgress** du composant **ChainingArgument**;
- le paramètre **exclusions** du composant **chainingArgument** (une liste de RDN à exclure de la recherche).

19.3.2.1.2 Résultats

Si cette procédure est correctement exécutée, elle retourne:

- un ensemble d'entrées mises en concordance et contenues dans le composant composite **searchResult.entryInformation**;
- **alreadySearched** dans **chainingResults**;
- un ensemble de références de continuation dans **SRcontinuationList**.

19.3.2.1.3 Erreurs

La procédure peut aboutir à l'une des erreurs suivantes, qui est alors retournée au DUA ou DSA:

- **accessControlError**;
- toute erreur définie pour la procédure **Find DSE** quand une entrée alias a été déréférencée.

19.3.2.2 Définition de la procédure

19.3.2.2.1 Procédure Search (I)

Il s'agit d'une procédure récursive applicable à une demande de recherche qui débute à une entrée donnée **e**. Elle recherche l'entrée cible **e** et ensuite les DSE immédiatement subordonnées à l'entrée **e**. La procédure s'invoque d'elle-même de manière récursive si la recherche porte sur un sous-arbre complet. Elle est constituée des étapes suivantes, représentées dans la Figure 19:

- 1) Si la DSE **e** est de type **cp** (c'est-à-dire une DSE avec préfixe de contexte) vérifier si un quelconque élément de l'argument **exclusions** est un préfixe du nom distinctif de **e**.
 - a) dans l'affirmative, retourner le résultat;
 - b) dans le cas contraire, appeler la procédure check Suitability
 - i) si l'entrée **e** ne convient pas, faire une référence **continuationReference** de la manière suivante et l'ajouter à **SRcontinuationList**:
 - **targetObject** mis au nom distinctif du supérieur immédiat de la DSE **e**;
 - **aliasedRDN** absents;
 - **operationProgress** avec **nameResolutionPhase** mis à **completed** et **nextRDNtoBeResolved** absent;
 - **rdnsResolved** absent;
 - **referenceType** mis à **supplier**;
 - **accessPoints** mis à **accessPointInformation** d'après la ou les valeurs trouvées dans l'attribut **supplierKnowledge** contenu dans **e**;Ensuite faire retour.

NOTE – C'est le seul endroit où une sous-demande de recherche (**nameResolutionPhase** est à **completed**) est chaînée à un fournisseur d'informations miroirs. Autrement dit, l'objet cible d'une telle sous-demande chaînée est toujours un préfixe de contexte.
 - ii) Par ailleurs, ajouter le nom DistinguishedName de **e** à **alreadySearched** dans **ChainingResults**;
- NOTE 1 – **alreadySearched** contient uniquement des préfixes de contexte.
- 2) Si **e** est du type **alias** et que **searchAliases** de **searchArguments** soit à **True**, appeler la procédure **Search Alias** et faire ensuite un retour;
- 3) Si **subset** est à **oneLevel**, passer à l'étape 6).

NOTE 2 – L'entrée **e** ne peut pas être une subordonnée incomplète à ce point étant donné que la procédure Check Suitability du préfixe de contexte a dû s'assurer que cela ne pouvait se produire.
- 4) Si **subset** est à **baseObject** ou si **entryOnly** est à **TRUE** et que de plus l'un des éléments suivants soit à **TRUE**:
 - a) **e** est du type **subentry** et la commande de service **subentry** est sélectionnée; ou

9594-4 : 1995 (F)

- b) **e** n'est pas du type **subentry** et la commande de service **subentry** n'est pas sélectionnée, effectuer les opérations suivantes:
- i) Vérifier l'information ACI. Si l'opération est interdite, passer à l'étape 6).
 - ii) Appliquer l'argument de filtre spécifié dans **SearchArgument.filter** à la DSE **e**. S'assurer que l'accès à tous les attributs utilisant les filtres est permis comme indiqué dans la Rec. UIT-T X.501 | ISO/CEI 9594-2. S'il y a concordance, ajouter les attributs sélectionnés par **SearchArgument.selection** à la liste des entrées qui correspondent dans **searchResult.entryInformation**. Ajouter uniquement les attributs qui ne sont pas plus grands que **attributeSizeLimit**. La façon dont les attributs collectifs appropriés sont traités lorsqu'on trouve une entrée dont les attributs sont à inclure dans un résultat relève des politiques locales.
 - iii) retour.
- Si a) et b) ne sont pas à true, passer à l'étape 6).
- 5) Si **subset** est à **subtree** (et si **entryOnly** n'est pas à **True**), et qu'en plus il soit vérifié:
- a) que l'entrée **e** est du type **subentry** et que la commande de service **subentry** est sélectionnée; ou
 - b) que l'entrée **e** n'est pas de type **subentry** et que la commande de service **subentry** n'est pas sélectionnée;
- procéder de la façon suivante:
- i) Vérifier l'information ACI. Si l'opération est interdite, passer à l'étape 6).
 - ii) Appliquer l'argument de filtre spécifié dans **searchArgument.filter** à la DSE **e**. S'assurer que l'accès à tous les attributs utilisant les filtres est permis, comme indiqué dans la Rec. UIT-T X.501 | ISO/CEI 9594-2. S'il y a concordance, ajouter les attributs sélectionnés par **searchArgument.selection** à la liste des entrées qui correspondent dans **searchResult.entryInformation**. La façon dont les attributs collectifs appropriés sont traités lorsqu'on trouve une entrée dont les attributs sont à inclure dans un résultat relève des politiques locales.
 - iii) Passer à l'étape 6).
- 6) Si **e** est de type **nssr**, ajouter une **continuationReference** à **SRcontinuationList** avec les composants suivants:
- **targetObject** mis au nom distinctif de la DSE **e**;
 - **aliasedRDN** absents;
 - **operationProgress** avec **nameResolutionPhase** mis à **completed** et **nextRDNtoBeResolved** absent;
 - **rdnsResolved** absent;
 - **referenceType** mis à **nssr**;
 - **accessPoints** mis à l'**accessPointInformation** tirée de la ou des valeurs trouvées dans l'attribut **nonSpecificKnowledge**.
- 7) Traiter toutes les sous-entrées DSE **e'** immédiatement subordonnées à l'entrée cible DSE **e** jusqu'à ce que toutes les DSE subordonnées aient été traitées. Si, au cours de cette boucle, la liste des entrées concordantes dans **searchResult.entryInformation** dépasse la taille limite, le temps limite ou la limite administrative, sélectionner le fanion **limit Problem** correspondant dans **partialOutcomeQualifier** et faire retour.

NOTE 3 – Le contrôle de la limite de taille est appliqué de manière implicite à chaque mise à jour de **searchResult**.

- a) Si la DSE **e'** est du type **subr** et non de type **cp**, ajouter une **continuationReference** à **SRcontinuationList** avec les composants suivants:
 - **targetObject** mis au nom de la DSE **e**;
 - **aliasedRDN** absents;
 - **operationProgress** avec **nameResolutionPhase** mis à **completed** et **nextRDNToBeResolved** absent;

- **rdnsResolved** absent;
- **referenceType** mis à **subr**;
- **accessPoints** mis à l'information relative aux points d'accès contenue dans l'attribut **specificKnowledge** de la DSE **e'**;

NOTE 4 – Si **e'** est à la fois de type **cp** et **subr**, une sous-demande de recherche pourra éventuellement être émise à partir de la référence subordonnée ou de la base de connaissance du fournisseur, mais non des deux. La présente procédure utilise la seconde solution (références de fournisseur trouvées dans le préfixe **cp**).

- b) Dans tous les autres cas, si la valeur du paramètre **subset** est **oneLevel**, mettre **entryOnly** à True et exécuter de manière récursive la procédure **Search Procedure(I)** pour l'entrée cible DSE **e'**.
- 8) Quand toutes les subordonnées ont été traitées, retourner à l'aiguilleur d'opérations pour la suite du traitement.

19.3.2.2.2 Procédure Search (II)

Cette procédure doit être suivie en cas de traitement d'une demande de recherche provenant d'une décomposition de demande au niveau du DSA dont on a reçu la demande. Cette procédure traite les DSE qui dépendent de l'entrée cible DSE **e** et appelle la procédure Search (I) pour chaque entrée d'objet.

- 1) Traiter toutes les sous-entrées DSE **e'** qui sont immédiatement subordonnées à l'entrée DSE **e** cible jusqu'à ce que toutes les DSE subordonnées aient été traitées. Lorsque toutes les entrées subordonnées ont été traitées, revenir à l'aiguilleur d'opérations pour la suite du traitement.
- 2) Si la DSE n'est pas de type **cp** et **entry**, l'ignorer et revenir à l'étape 1).
- 3) Si la DSE est de type **cp** et **entry**, appeler la procédure **Check Suitability**. Si le résultat est favorable, passer à l'étape 4). Sinon, ignorer cette entrée et passer à l'étape 1).
- 4) Exécuter la procédure **Search (I)** pour la DSE **e'** comme indiqué au 19.3.2.2. Si la DSE est du type **alias** et que la valeur du paramètre **subset** est à **oneLevel**, mettre le composant **chainingArgument.entryOnly** à True à l'appel de la procédure **Search (I)**. Revenir à l'étape 1).

19.3.2.2.3 Procédure Search Alias

Cette procédure est exécutée si une DSE de type **alias** a été rencontrée au cours du traitement d'une demande de recherche.

- 1) Si le paramètre **subset** a la valeur **baseObject** ou **oneLevel**, passer à l'étape 4).
- 2) Si le nom **aliasedEntryName** est un préfixe de **targetObject** ou de **baseObject**, l'alias est exclu de la recherche parce que cela entraînerait une recherche récursive produisant des résultats en double;
- 3) Si **targetObject** ou **baseObject** est un préfixe du nom **aliasedEntryName**, aucun traitement spécifique de l'alias n'est requis étant donné que le sous-arbre d'alias sera recherché quoi qu'il arrive.

NOTE – Dans les deux cas ci-dessus, **baseObject** peut ne pas être un préfixe de **targetObject** en raison du déréférencement d'alias.

- 4) Etablir une demande de protocole **DSP** avec l'objet cible mis au nom **aliasedEntryName**. Si le paramètre **subset** est à **oneLevel**, mettre **entryOnly** à **True**. Appeler l'aiguilleur d'opérations pour que la demande soit continuée localement.
- 5) Si l'aiguilleur d'opérations retourne une erreur de renvoi de référence ou d'occupation ou d'indisponibilité, ajouter (ou constituer et ajouter) la référence de continuation au qualificateur **partialOutcomeQualifier** de **searchResult**, puis retourner le résultat.
- 6) Si l'aiguilleur d'opérations retourne d'autres erreurs, les rejeter et retourner le résultat;
- 7) Si l'aiguilleur d'opérations retourne un résultat **searchResult**,
 - i) si ce résultat est signé, l'ajouter aux informations **uncorrelatedSearchInfo** contenues dans le résultat **SearchResult**;
 - ii) si ce résultat n'est pas signé, l'ajouter aux informations **searchInfo** contenues dans le résultat **searchResult** et retourner l'ensemble.

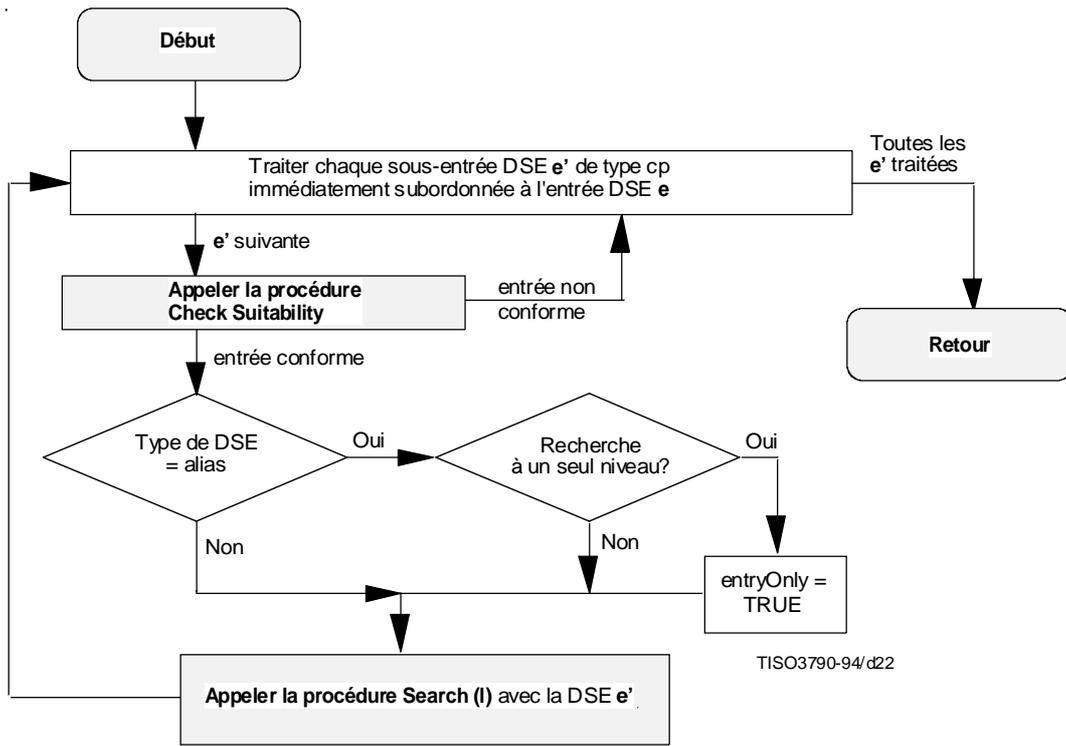


Figure 20 – Procédure Search (II)

20 Procédures de référence de continuation

Les procédures décrites dans le présent article sont appelées pour traiter la liste des références de continuation (**NRcontinuationList** ou **SRcontinuationList**) créées par d'autres procédures.

Les procédures de référence de continuation comprennent les étapes présentées dans les Figures 24, 25 et 26. La première étape consiste à identifier, dans la liste de continuation, des ensembles de références de continuation qui ont un composant d'objet cible commun. Ces ensembles de références ont été créés à partir d'un ensemble de références subordonnées ou de références subordonnées non spécifiques associées à la même entrée dans le DIT.

Ces ensembles (chacun avec un composant d'objet cible différent) peuvent être traités indépendamment, successivement ou parallèlement par le DSA, étant donné qu'il n'y a aucun risque que les mêmes résultats soient retournés par deux ensembles quelconques. Toutefois, le traitement de chaque référence de continuation dans un ensemble, de chaque information de point d'accès dans une référence de continuation et de chaque point d'accès dans une information de point d'accès (API) doit être contrôlé; sinon on peut obtenir les résultats en double, comme décrit au 20.1.

La procédure APIInfo consiste à traiter un par un tous les points d'accès de l'ensemble contenu dans une information de point d'accès unique. Tous ces points désignent l'original (ou des copies) du même contexte de dénomination (ou éventuellement un ensemble de contextes de dénomination détenus par un même DSA dans le cas de NSSR). Si le premier point d'accès aboutit à un résultat ou à une erreur permanente, les autres points d'accès n'ont normalement pas besoin d'être traités. Toutefois, si l'erreur est une erreur temporaire, c'est-à-dire une erreur de service (**busy**, **unavailable**, **unwillingToPerform**, **invalidReference** ou **administrativeLimitExceeded**), le DSA peut choisir, à titre d'option locale, de traiter un autre point d'accès de l'ensemble.

9594-4 : 1995 (F)

Le traitement des valeurs d'information de point d'accès dans un ensemble de références de continuation est effectué de façon uniforme, quelle que soit la référence de continuation dont elles proviennent. (Explication: deux DSE de type **subr** dépendant d'une entrée unique donneront deux références de continuation, chacune contenant une seule valeur **accessPoint Information**, alors qu'une seule DSE de type référence NSSR aux deux mêmes objets subordonnés (à supposer qu'ils soient détenus par des DSA différents) donnerait une référence de continuation contenant un ensemble de deux valeurs **accessPointInformation**).

Les valeurs de type **accessPointInformation** peuvent être traitées successivement ou parallèlement, comme décrit au 20.1. Il y a plus de risques d'obtenir des résultats en double avec la stratégie parallèle. Il faut toujours supprimer les résultats en double.

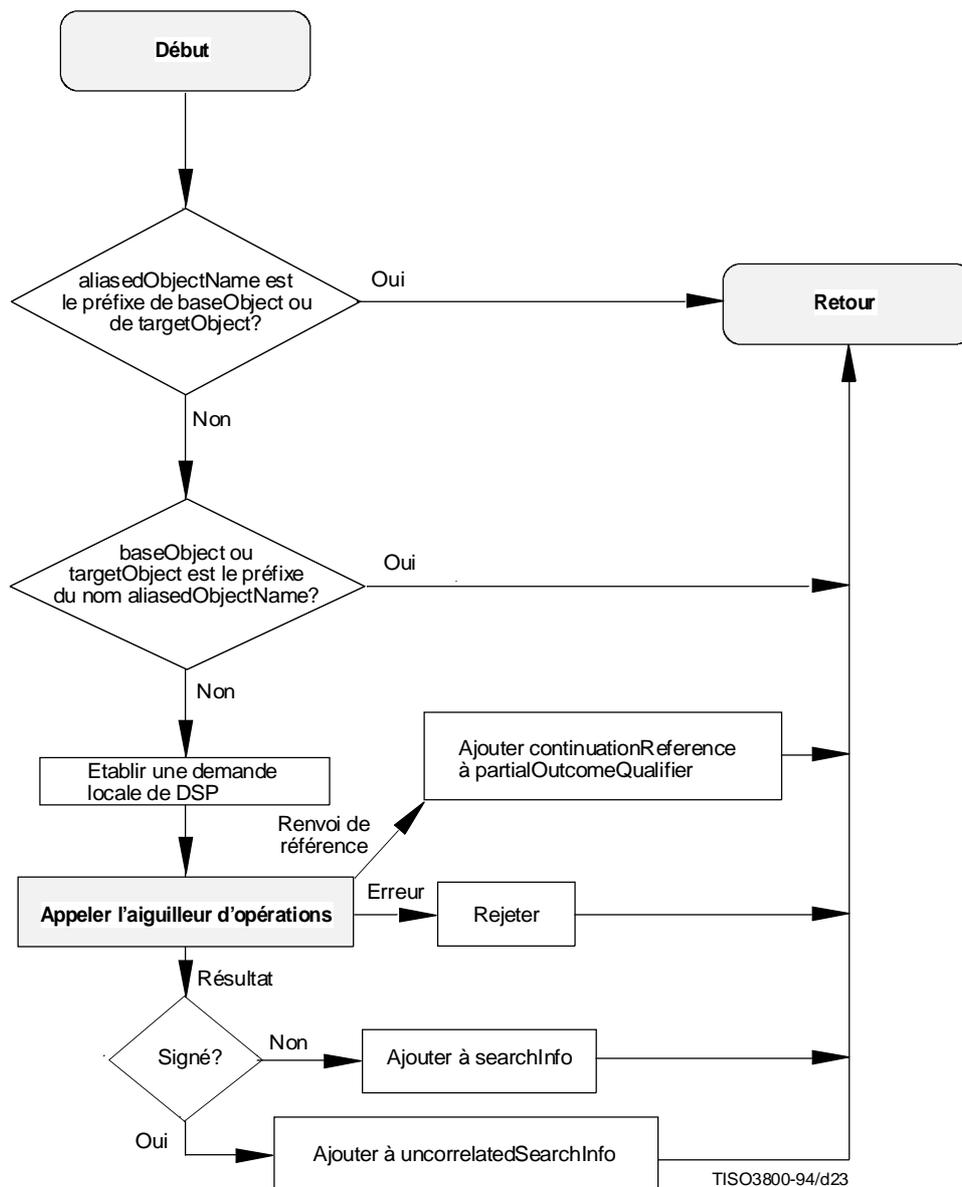


Figure 21 – Procédure Search alias

20.1 Stratégie de chaînage en présence de duplication miroir

En présence de duplication miroir, un DSA peut choisir entre différentes stratégies lorsqu'il doit multichaîner une demande à plusieurs DSA. Ce choix se présente toujours si le DSA doit traiter plusieurs références de continuation avec le même **targetObject**. Cette situation peut découler du multichaînage causé par décomposition de NSSR pendant la résolution du nom (voir la Figure 22) ou d'une décomposition de demande pendant l'évaluation d'une opération portant sur plusieurs objets (Figure 23).

Le but de ces stratégies est de traiter le problème des résultats obtenus en double et du traitement en double lorsque des informations miroirs sont utilisées dans un multichaînage de demandes (causé par décomposition de NSSR ou de demande). Par exemple dans la Figure 22, le DSA 1 multichaîne une demande à la fois vers les DSA 2 et 3 car la NSSR est détenue dans la DSE B. Si l'utilisation d'informations miroirs est autorisée, les DSA 2 et 3 peuvent appliquer l'opération chaînée aux deux sous-arbres commençant aux points X et Y.

De même, dans la Figure 23, le DSA 1 multichaîne une sous-demande (à la suite d'une décomposition de demande) aux deux références subordonnées détenues aux DSE X et Y. De nouveau, si l'utilisation d'informations miroirs est autorisée, les DSA 2 et 3 peuvent tous deux appliquer l'opération chaînée aux deux sous-arbres en commençant aux points X et Y.

Pour résoudre ce problème de duplication miroir, un DSA peut choisir l'une des stratégies suivantes en cas de multichaînage à des demandes de DSA multiples avec le même **targetObject**.

20.1.1 La stratégie maître seulement

Un DSA peut choisir cette stratégie pour éviter d'utiliser des informations miroirs lorsqu'il exécute un multichaînage parallèle ou séquentiel causé par une décomposition de nssr ou par une décomposition de demande pendant une évaluation d'opération de recherche ou de liste. Pour cette stratégie, pendant une évaluation d'opération de recherche ou de liste, le composant **excludeShadows** de **ChainingArgument** est mis à **TRUE**. Si des NSSR sont rencontrées pendant la résolution du nom, un DSA peut mettre **nameResolveOnMaster** à **TRUE** pour s'assurer que seul un chemin unique est suivi. **nameResolveOnMaster** doit être mis à **TRUE** si des NSSR sont rencontrées et que l'opération soit l'une des opérations de modification de l'Annuaire. Dans chaque cas, seuls le ou les DSA qui détiennent l'entrée (ou les entrées) du maître concernant l'opération doivent effectuer cette opération. Cette stratégie maître seulement peut être utilisée pendant le multichaînage aussi bien parallèle que séquentiel.

20.1.2 La stratégie parallèle

Lorsqu'il utilise cette stratégie, un DSA envoie toutes les demandes chaînées sous forme de multichaînage parallèle. Cette stratégie peut être utilisée pendant l'évaluation de recherche ou de listage et la résolution du nom des NSSR. Cela permettra d'utiliser des informations miroirs pour traiter les demandes chaînées, mais cela risque d'entraîner une duplication des exécutions et des résultats pour l'opération. Si un DSA choisit cette stratégie, il devra supprimer du résultat de l'opération qu'il retourne les résultats obtenus en double.

Etant donné qu'il n'est pas possible de supprimer des résultats obtenus en double lorsqu'un résultat signé a été demandé, un DSA ne devra pas choisir cette stratégie si des résultats signés sont demandés pendant l'évaluation de la recherche, à moins que le composant **excludeShadow** ne soit aussi sélectionné.

20.1.3 La stratégie séquentielle

Cette stratégie évite d'obtenir des résultats en double grâce à l'utilisation du multichaînage séquentiel pour traiter les (sous)-demandes chaînées d'une décomposition de recherche ou de NSSR. Chaque demande chaînée est traitée séparément.

S'agissant de la décomposition d'une NSSR, si un résultat ou une erreur permanente fait suite à une demande, il n'est pas nécessaire de chaîner les autres demandes. Si une erreur temporaire est renvoyée, une demande ultérieure peut être chaînée, ou bien l'erreur temporaire peut être renvoyée au demandeur, selon la politique locale.

S'agissant d'une évaluation de recherche, le composant **exclusions** des **chainingArgument** est mis sur l'ensemble des RDN qui ont déjà été traités. Pour cela, on incorpore les éléments de **chainingResult.alreadySearched** dans l'argument **exclusions** de la demande chaînée suivante. C'est la seule stratégie qui évite complètement la duplication pendant l'évaluation de recherche.

Une stratégie séquentielle n'est pas définie pour l'évaluation de listage (bien que le multichaînage séquentiel puisse être utilisé), étant donné qu'un DSA supérieur ne peut pas empêcher que des subordonnés spécifiques soient renvoyés dans des sous-demandes de liste ultérieures (à noter que le composant **excludeShadows** n'exclut pas de subordonnés spécifiques mais est un moyen rapide d'exclure toutes les duplications miroirs).

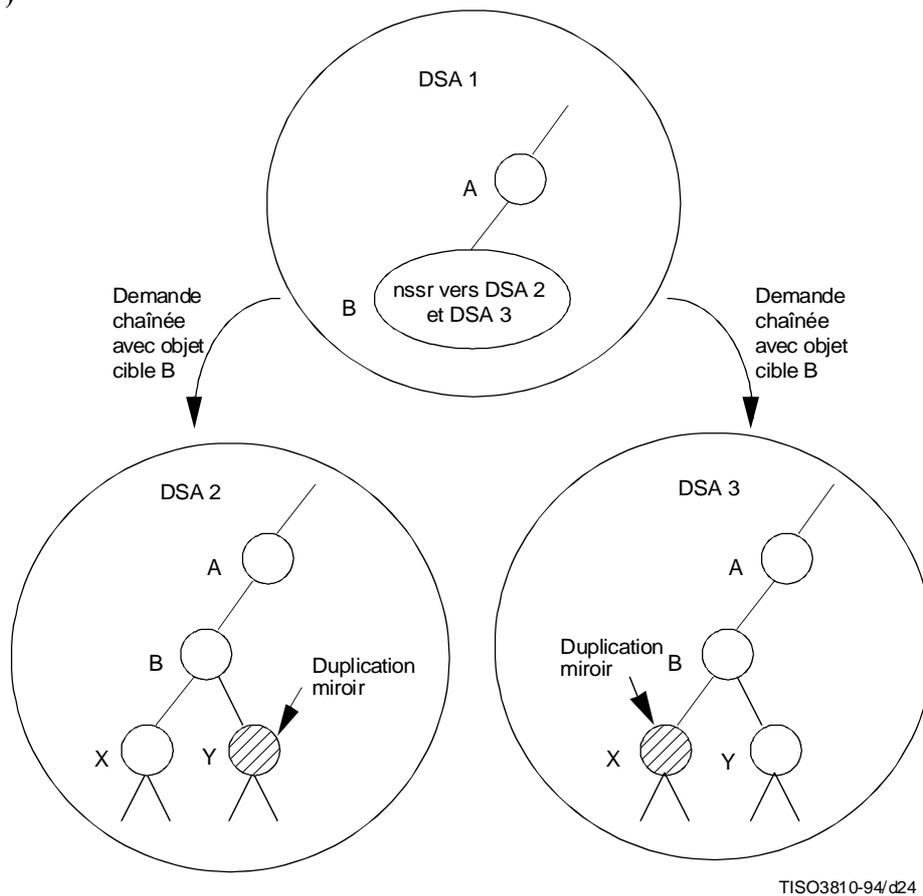


Figure 22 – Multichainage causé par une NSSR pendant une résolution du nom

20.2 Emission de sous-demandes chaînées vers un DSA distant

Avant l'émission d'une sous-demande, un DSA doit exécuter une opération de rattachement **BIND** lorsqu'il doit établir une association avec le DSA distant. La gestion des associations n'entre pas dans le cadre des présentes Spécifications d'Annuaire. Une association avec un autre DSA est considérée comme non disponible si l'association ne peut être établie ou si le DSA, pour des raisons locales, décide de ne pas établir d'association. En pareil cas, l'opération **BIND** a échoué. C'est au niveau local que l'on décide du moment où l'on cesse d'essayer d'établir une association et où l'on déclare que l'association n'est pas disponible.

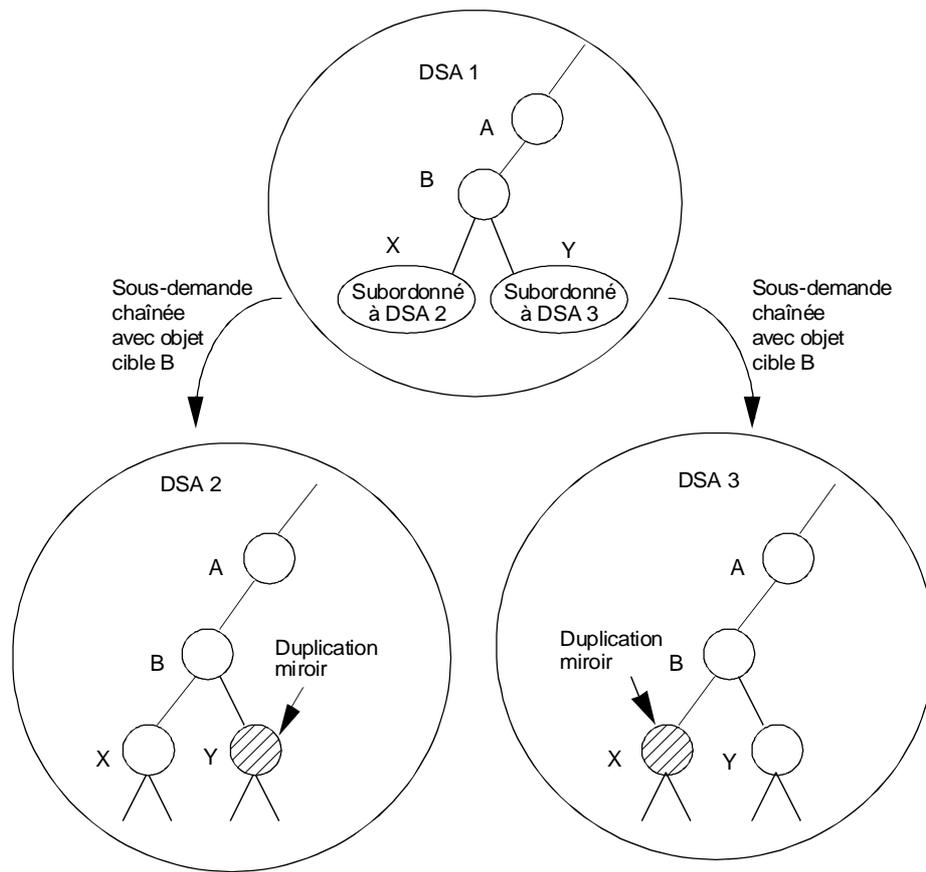
Lorsqu'un DSA essaie l'opération de rattachement **BIND** pour établir une association avec un autre DSA et qu'il reçoit le message **BindError**, l'émission de la sous-demande a échoué.

20.3 Paramètres des procédures

20.3.1 Arguments

Ces procédures utilisent les arguments suivants:

- la liste des références de continuation à traiter dans **NRcontinuationList** (pour la procédure de Référence de continuation de résolution du nom) et **SRcontinuationList** (respectivement pour la procédure de référence de continuation de listage et pour celle de recherche);
- le composant **commonArguments** de l'argument d'opération;
- le composant **chainingArgument**.



TISO3820-94/d25

Figure 23 – Décomposition d'une demande de multichaînage à l'aide de références subordonnées

20.3.2 Résultats

Ces procédures créent les résultats suivants:

- une liste des messages de résultats/erreurs reçus à la suite des demandes chaînées émises, si le chaînage a été choisi;
- une liste mise à jour des références de continuation non traitées, contenues dans **continuationList**.

20.3.3 Erreurs

Ces procédures peuvent retourner l'une des erreurs suivantes:

- **serviceError** avec **outOfScope** si un renvoi de référence n'appartenant pas au **scopeOfReferral**, a été créé;
- **serviceError** avec **DITerror** lorsqu'une référence de connaissance non valide a été détectée;
- **nameError** avec **noSuchObject** lorsque toutes les sous-demandes issues de la décomposition de NSSR ont retourné l'erreur **unabletoProceed**;
- toute autre erreur qui est retournée par une sous-demande chaînée;
- **referralError** lorsque le chaînage n'a pas été choisi et que **operationProgress.nameResolution** est mis à **notStarted** ou à **proceeding**.

9594-4 : 1995 (F)

20.4 Définition des procédures

Si le composant composite **operationProgress.nameResolutionPhase** est mis à **notStarted** ou à **proceeding**, la procédure décrite au 20.4.1 (procédure de référence de continuation de résolution du nom) doit être suivie. Les opérations d'interrogation à entrées multiples, listage et recherche, appellent respectivement les procédures décrites aux 20.4.2 et 20.4.3.

20.4.1 Procédure de référence de continuation pour la résolution du nom

Cette procédure comprend les étapes présentées à la Figure 24. Le principe de base de cette procédure consiste à traiter successivement l'ensemble des références de continuation créées pendant la résolution du nom.

Les étapes suivantes doivent être exécutées pour chaque référence de continuation **C** contenue dans la **NRcontinuationList** dans un ordre donné jusqu'à ce que toutes les références aient été traitées ou qu'une erreur ou un résultat ait été retourné. Si toutes les références ont été traitées, retourner à **Operation Dispatcher** pour poursuivre la procédure **Result Merging** en vue de traiter le résultat ou le renvoi de référence reçu.

- 1) Vérifier si **chainingProhibited** est sélectionné. S'il est sélectionné, le DSA n'est pas autorisé à chaîner. Selon la politique locale (**local policy**), une **ServiceError** avec la cause **chainingRequired** ou un **Referral** est retourné à **Operation Dispatcher**.
- 2) Si **ChainingProhibited** n'est pas sélectionné, vérifier si la politique locale (**local policy**) permet le chaînage. Si le chaînage n'est pas autorisé, retourner un **Referral**. Si la politique locale (**local policy**) permet le chaînage, passer à l'étape suivante.
- 3) Traiter chacune des références de continuation de la liste des références de continuation figurant dans **NRcontinuationList**. S'il n'y a plus de références de continuation non traitées, retourner un message **ServiceError**.
- 4) Traiter la référence de continuation **C** suivante de la **NRcontinuationList**. S'il s'agit d'une **NSSR**, passer à l'étape 5). S'il ne s'agit pas d'une **NSSR**, appeler la procédure **APIInfo** pour la traiter. Il convient de faire une distinction entre les retours possibles par la procédure **APIInfo**:
 - Si la procédure **APIInfo** retourne **null result**, revenir à l'étape 3) et traiter la référence de continuation suivante.
 - Si **APIInfo procedure** retourne **error**, **Referral** or **result**, le renvoyer.
- 5) En pareil cas, la référence de continuation est de type **NSSR** et le DSA a le choix entre un chaînage séquentiel ou parallèle, selon le choix local de stratégie (**local choice of strategy**). Si la **NSSR** doit être **processed sequentially**, passer à l'étape 6). Si elle doit être **processed in parallel**, alors pour chaque **AccessPointInformation (API)** de la **NSSR**, **APIInfo procedure** est appelée pour que ces informations soient traitées en parallèle. Attendre que toutes les **API** soient traitées, c'est-à-dire attendre que tous les appels à **APIInfo procedure** soient retournés. Vérifier tous les résultats provenant de ces appels dans l'ordre suivant:
 - Si tous les appels retournent **ServiceError** avec la cause **unableToProceed**, retourner **NameError**.
 - Si l'on reçoit un ou plusieurs **results**, éliminer les duplications éventuelles et retourner le **result**.
 - Si l'on reçoit une **Error** et qu'il n'est pas une **ServiceError** (par exemple: **NameError**), retourner **error**.
 - Dans les autres cas, retourner **Referral** ou **ServiceError** vers **Operation Dispatcher**, selon le choix local.
- 6) Choisir l'**API** non traitée suivante parmi l'ensemble des **API** de la **NSSR** et passer à l'étape 7). Si toutes les **API** ont été traitées, vérifier que tous les appels à la procédure **APIInfo** ont retourné le message **ServiceError** avec la cause **unableToProceed**. Si tel est le cas, l'entrée ne peut pas être trouvée et **NameError** est retourné; sinon, selon le choix local, **Referral** ou **ServiceError** est retourné.
- 7) Appeler **APIInfo procedure**. Il convient de faire une distinction entre les résultats possibles des appels à **APIInfo procedure**:
 - Si l'on reçoit **ServiceError** avec la cause **unableToProceed**, essayer un autre point d'accès. Revenir à l'étape 6).

- Si l'on reçoit **ServiceError** avec la cause **busy**, **unavailable**, **unwillingToPerform** ou **invalidReference**, la cause indiquée peut être de nature transitoire et c'est au niveau local que l'on choisit d'essayer de chaîner la demande vers un autre DSA. Si l'on choisit d'essayer un autre DSA, revenir à l'étape 6); sinon, retourner **Referral** ou **ServiceError**, selon le choix local.
- Si l'on reçoit une erreur autre que **ServiceError** avec la cause **busy**, **unavailable**, **unwillingToPerform**, **invalidReference** ou **unableToProceed**, cette erreur doit être retournée à **Operation Dispatcher**. Si **ServiceError** est **invalidReference**, une conversion en **DITError** est nécessaire avant le renvoi au demandeur.
- Si l'on reçoit **result** ou **Referral**, le renvoyer à **Operation Dispatcher**.

20.4.3 Procédure de référence de continuation pour le listage

Cette procédure comprend les étapes indiquées à la Figure 25. Elle est invoquée lorsqu'une demande de listage ne peut être satisfaite dans le DSA local et qu'un ensemble de références de continuation a été ajouté à **SRcontinuationList** pour le chaînage ou le renvoi de référence. Toutes ces références de continuation (CR) ont le même **targetObject**. Les CR avec **referenceType nssr** ont une ou plusieurs valeurs **accessPointInformation** (API), alors que d'autres types de CR ont une seule API. Chacune de ces API est extraite et examinée en vue du chaînage ou du renvoi de référence.

Il convient de suivre les étapes suivantes:

- 1) Si une quelconque cause d'erreur par dépassement de limite a été détectée d'ici là, revenir à l'aiguilleur **Operation Dispatcher** pour poursuivre la procédure **Result Merging**.
- 2) Si le fanion **chainingProhibited** dans **commonArguments.serviceControls** est sélectionné ou si le DSA décide de ne pas faire de chaînage en raison de sa politique d'exploitation locale, le DSA doit revenir directement à l'aiguilleur **Operation Dispatcher** pour poursuivre la procédure de fusionnement **Result Merging**.
- 3) Créer un ensemble de valeurs **AccessPointInformation** à partir du composant **accessPoints** de chaque référence de continuation de la **SRcontinuationList**.

Utiliser la stratégie parallèle ou séquentielle pour traiter chacune des API de la façon suivante:

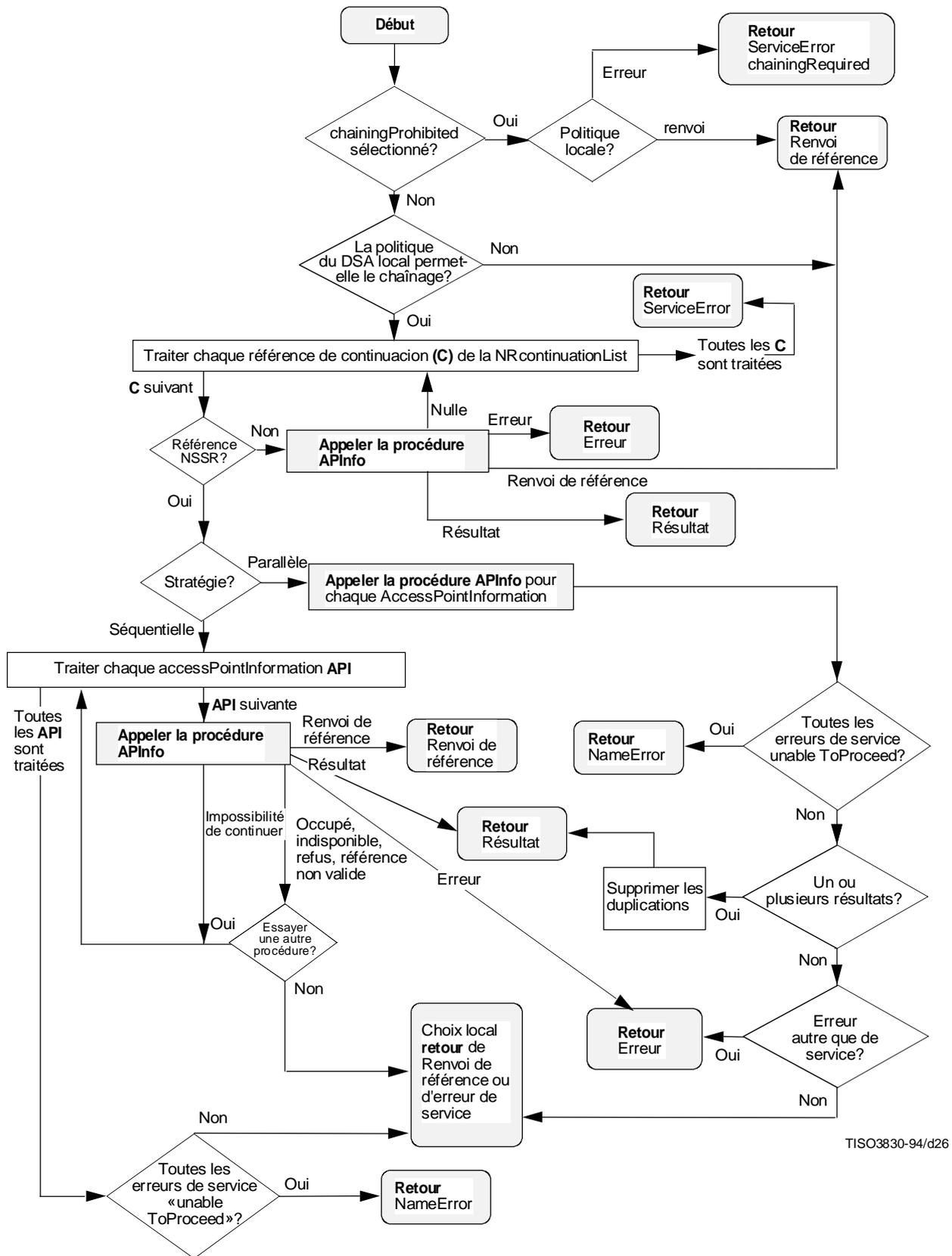
- i) Appeler la procédure **APIInfo** avec l'information API suivante de l'ensemble;
- ii) Si un résultat est retourné, l'ajouter à **listInfo** s'il n'est pas signé ou l'ajouter à **uncorrelatedListInfo** s'il est signé.
- iii) Si le retour est une erreur ou un résultat nul, cela signifie que la procédure **APIInfo** a déjà essayé tous les points d'accès de l'API sans succès. Selon la politique de sécurité et d'exploitation locale, ne pas en tenir compte et passer à l'API suivante ou ajouter une référence de continuation fondée sur cette API à **partialOutcomeQualifier**.

NOTE – Il n'est pas plausible d'obtenir un renvoi de référence par la procédure **APIInfo**. Tout «renvoi de référence» devrait se présenter sous la forme **unexplored** dans le qualificateur **partialOutcomeQualifier**.

- 4) Lorsque toutes les API ont été traitées, revenir à l'aiguilleur **Operation Dispatcher**.

20.4.4 Procédure de référence de continuation pour la recherche

Cette procédure comprend les étapes indiquées à la Figure 26. Elle est invoquée lorsqu'une demande de recherche ne peut être satisfaite dans le DSA local et qu'un ensemble de références de continuation a été ajouté à **SRcontinuationList** pour le chaînage ou le renvoi de référence. La procédure ressemble beaucoup à la **List Continuation Reference procedure**. La différence est que, dans ce cas, les références de continuation figurant dans **SRcontinuationList** peuvent avoir différentes valeurs du **targetObject**. En conséquence, les références de continuation sont classées en ensembles de références de continuation ayant le même **targetObject**. De plus, l'utilisation de réponse et d'argument de chaînage **exclusions** et **alreadySearched** est définie, car il s'agit d'une stratégie importante pour la recherche. L'utilisation de **exclusions** et **alreadySearched** s'applique au traitement de chaque ensemble de références de continuation ayant le même **targetObject**.



TISO3830-94/d26

Figure 24 – Procédure de référence de continuation pour la résolution du nom

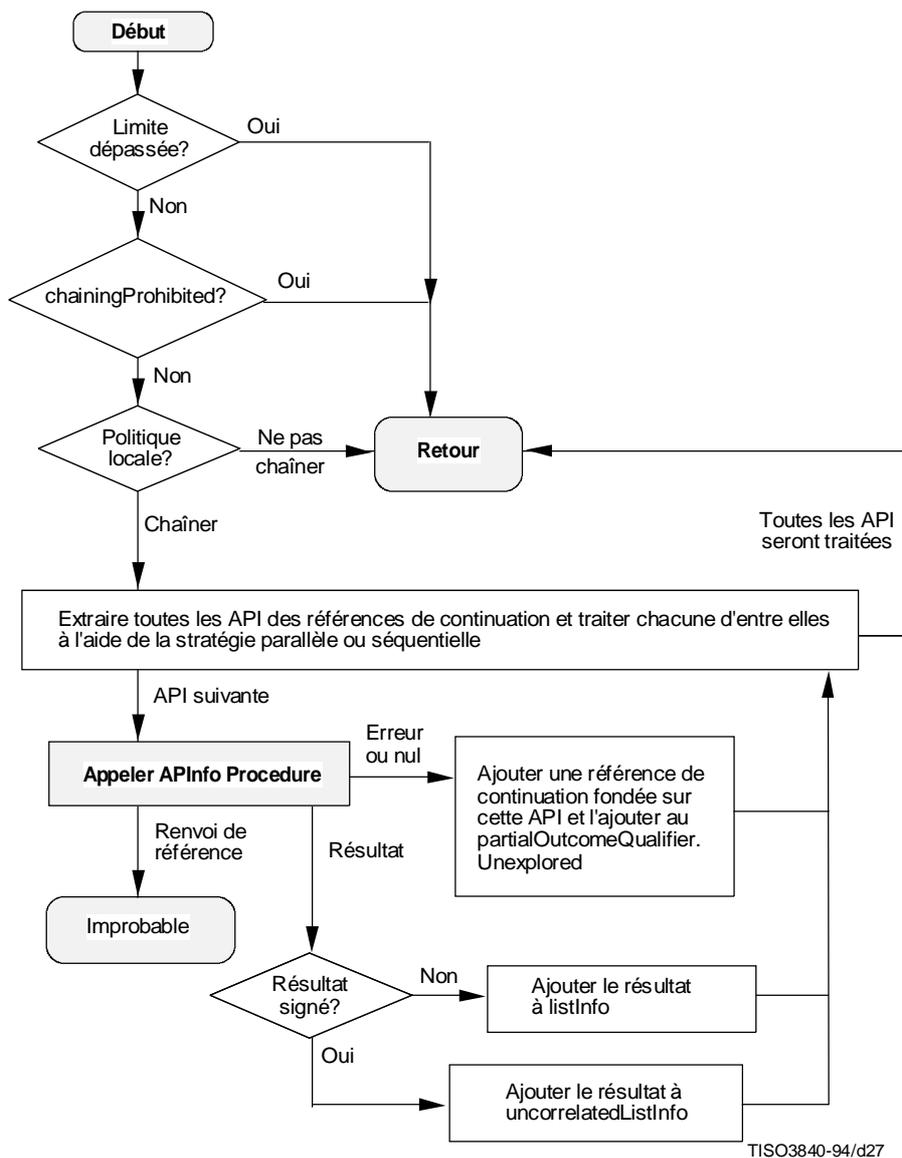
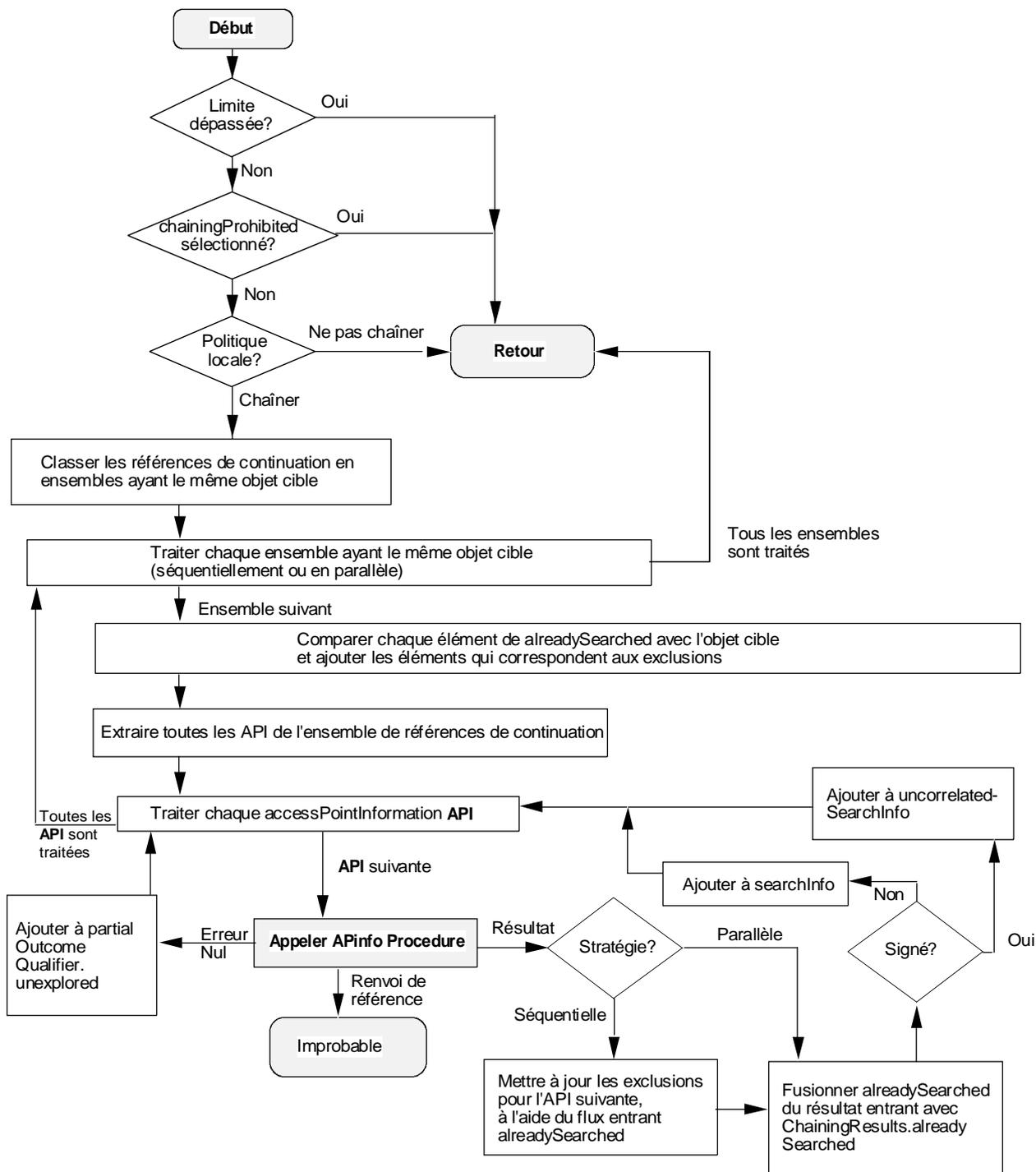


Figure 25 – Procédure de référence de continuation pour le listage



TISO3850-94/d28

Figure 26 – Procédure de référence de continuation pour la recherche

Il convient de suivre les étapes suivantes:

- 1) Si la limite a été dépassée, revenir à l'aiguilleur **Operation Dispatcher** pour passer à la procédure **Result Merging**.
- 2) Si le fanion **chainingProhibited** dans **commonArguments.serviceControls** est sélectionné ou si le DSA décide de ne pas effectuer de chaînage en raison de sa politique d'exploitation locale, le DSA reviendra directement à l'aiguilleur **Operation Dispatcher** pour passer à la procédure **Result Merging**.
- 3) Classer les références de continuation figurant dans **SRcontinuationList** en ensembles ayant le même **targetObject**.
- 4) Pour chaque sous-ensemble de références de continuation, créer un ensemble de valeurs **AccessPointInformation** à partir du composant **accessPoints** de chaque référence de continuation du sous-ensemble et choisir la stratégie séquentielle ou parallèle pour poursuivre le traitement. Si l'on choisit la stratégie parallèle, sauter les étapes indiquées ci-dessous, qui ne s'appliquent qu'à la stratégie séquentielle.
 - a) Si l'on choisit la stratégie séquentielle, enregistrer une variable locale **localExclusions** pour chaque ensemble de références de continuation ayant le même **targetObject**. Au départ, la variable **localExclusions** est mise sur les **exclusions** de la demande de chaînage entrante (le cas échéant) et de tous les sous-arbres recherchés au niveau local qui dépendent directement du **targetObject**.
 - b) Si l'on utilise la stratégie séquentielle, comparer le **targetObject** à tous les éléments de la variable **localExclusions** et supprimer les éléments qui ne contiennent pas comme préfixe **targetObject**. On obtient ainsi les exclusions pertinentes pour l'objet cible actuel.
 - c) Extraire toutes les API de l'ensemble des références de continuation visant l'objet cible actuel.
 - d) Effectuer l'itération de chaque API. Pour chaque API:
 - i) Appeler la procédure **APIInfo**.
 - ii) Si un résultat est retourné, ajouter le résultat au composant **searchInfo** s'il n'est pas signé, ou l'ajouter au composant **uncorrelatedSearchInfo** s'il est signé. Si l'on utilise la stratégie séquentielle, mettre à jour la variable **localExclusions** à l'aide de l'élément **alreadySearched** dans la réponse entrante et fusionner **alreadySearched** avec la réponse entrant au composant **chainingResult.alreadySearched** du DSA. Passer ensuite à l'API suivante.
 - iii) Si une erreur ou un résultat nul est retourné, cela signifie que la procédure **APIInfo** a déjà essayé tous les points d'accès de l'API sans succès. Selon la politique de sécurité et d'exploitation locale, ne pas en tenir compte et passer à l'API suivante ou ajouter une référence de continuation fondée sur cette API à **partialOutcomeQualifier**.

NOTE – Il n'est pas plausible d'obtenir un renvoi de référence par la procédure **APIInfo**. Tout «renvoi de référence» devrait se présenter sous la forme de l'élément **unexplored** dans le **partialOutcomeQualifier**.
 - e) Lorsque toutes les API ont été traitées, passer à l'ensemble suivant de références de continuation ayant le même **targetObject**.
- 5) Lorsque toutes les références de continuation sont traitées, revenir à l'aiguilleur à **Operation Dispatcher**.

20.4.5 Procédure **APIInfo**

Cette procédure est appelée pour traiter une API (**accessPointInformation**), qui contient un ou plusieurs points d'accès. Les points sont traités un par un jusqu'à ce qu'un résultat ou une erreur soit retourné. Si l'erreur est une erreur de service qui permette d'essayer un autre point avec succès, on essaie d'autres points d'accès dans la mesure où la politique d'exploitation locale le permet.

- 1) Procéder à la détection de boucle. Si une boucle est détectée, retourner le message **ServiceError** avec la cause **loopDetected**. Sinon, passer à l'étape 2).
- 2) Traiter chacun des points d'accès (AP) de l'information de point d'accès (API). Si tous les points ont été traités, retourner **null result**. S'il y a un point d'accès à traiter, passer à l'étape 3).
- 3) Vérifier si la politique locale permet le chaînage à ce point d'accès. Cette vérification doit tenir compte des réglages des commandes de service et des arguments de chaînage (par exemple: **chainingProhibited**, **preferChaining**, que le point d'accès soit ou non dans **localScope**, **excludeShadows**. Si la politique locale ou le réglage des commandes de service respectives ne permet pas d'utiliser ce point d'accès donné, ne pas tenir compte du point d'accès et passer à l'étape 2). Si le point d'accès peut être utilisé, passer à l'étape 4).

9594-4 : 1995 (F)

- 4) Si la politique locale a conduit à choisir la stratégie maître seulement, mettre l'argument de chaînage **excludeShadows** à **True**.
- Si **nameResolutionPhase** n'est pas **completed** et si la stratégie doit continuer la résolution du nom au sujet des entrées du maître, mettre **nameResolveOnMaster** sur **True**.
- L'argument de chaînage **nameResolveOnMaster** est mis sur **True** si l'un des cas suivants se présente:
- dans l'argument de chaînage entrant, **nameResolutionPhase** se poursuit et **nameResolveOnMaster** est mis sur **True**; ou
 - l'opération est l'une des opérations de modification, le **referenceType** de la demande de chaînage à émettre est une NSSR, et l'on utilise une stratégie parallèle.
- NOTE – L'utilisation du composant **nameResolveOnMaster** vise à empêcher d'effectuer plusieurs fois les opérations de modification en raison de la présence d'une NSSR.
- 5) Etablir une demande chaînée et essayer de l'émettre:
- a) Appliquer la procédure de prévention de boucle en vérifiant si un item ayant le même **targetObject** et **operationProgress** apparaît dans **traceInformation** du **chainingArgument** reçu. Si la demande qui en résulte (comme décrit à l'étape 5), b) aboutit à une boucle, le DSA doit retourner le message **ServiceError** avec la cause **loopDetected** au DUA/DSA demandeur ou ne pas tenir compte du point d'accès et essayer le point d'accès suivant en revenant à l'étape 2).
 - b) A la suite d'un rattachement d'association réussi, le DSA doit émettre une opération chaînée du même type que l'opération qui est traitée avec les paramètres suivants:
 - l'argument d'opération dans l'opération chaînée, mis sur la même valeur que l'argument d'opération reçu;
 - **chainingArguments.originator**, mis à la valeur reçue;
 - **chainingArguments.targetObject** mis au **targetObject** de la référence de continuation;
 - **chainingArguments.operationProgress** mis à la valeur de **operationProgress** de la référence de continuation
 - **chainingArguments.traceInformation** mis à l'information de trace, telle qu'elle est mise à jour par la procédure **Request Validation**;
 - **chainingArguments.aliasDereferenced** mis à la valeur actualisée de **aliasDereferenced** mis à jour au niveau local;
 - **chainingArguments.returnCrossReferences** mis à choix local;
 - **chainingArguments.referenceType** mis à la valeur du **referenceType** de la référence de continuation;
 - **chainingArguments.timeLimit** mis à la valeur de **timeLimit** reçue;
 - **chainingArguments.exclusions** absent;
 - **SecurityParameters** mis à la valeur des **SecurityParameters** reçus.
- 6) Si la demande n'a pas pu aboutir, passer à l'étape 7); si elle a pu aboutir, passer à l'étape 8).
- 7) On décide au niveau local de continuer ou pas. Si le DSA choisit de continuer, on ne tient pas compte de l'erreur et l'on essaie le point d'accès suivant. Revenir à l'étape 2). Si le DSA décide de ne pas essayer d'autre point d'accès, on décidera, selon la politique locale, de retourner ou non un message **Referral** ou **ServiceError**, selon le cas, au demandeur de la procédure.
- 8) Si la demande a pu aboutir, le DSA attend la réponse et la traite:
- a) Si l'on reçoit un résultat **result**, celui-ci est retourné au demandeur de la procédure.
 - b) Si l'on reçoit l'erreur **ServiceError** avec la cause **busy**, **unavailable**, **unwillingToPerform** ou **invalidReference**, revenir à l'étape 7).
 - c) Si l'on reçoit le renvoi **Referral** et que **returnToDUA** est mis sur **TRUE**, le DSA qui reçoit ne donne pas suite au renvoi, mais le retourne au demandeur.
 - d) Si l'on reçoit le renvoi **Referral** et que **returnToDUA** est mis sur **FALSE**, il convient d'appliquer les mêmes considérations de politique locale qu'à l'étape 3) (compte tenu des commandes de service, des arguments de chaînage, de la stratégie de chaînage, etc.). Si l'on décide de ne pas déréférencer le

renvoi Referral, retourner celui-ci au demandeur. S'il est décidé de référencer le renvoi Referral, vider la liste **NRcontinuationList**, placer la référence de continuation (telle qu'elle a été reçue dans la liste renvoi Referral) dans **NRcontinuationList** et appeler la procédure Name Resolution Continuation Reference. Cela peut aboutir à un **result**, un **Referral**, une **ServiceError** ou à une autre **error**. Quel que soit le résultat de l'appel de la procédure Name Resolution Continuation Reference, ce résultat sera communiqué au demandeur.

- e) Si une autre **error** se produit, elle sera renvoyée au demandeur.

20.5 Procédure d'abandon

Cette procédure est invoquée si l'on reçoit une demande d'abandon. Elle comprend les étapes suivantes, indiquées à la Figure 28.

- 1) Lorsque l'on reçoit une demande d'abandon qui se réfère à une opération inconnue, il faut retourner au demandeur une erreur **AbandonError** avec une valeur **noSuchOperation**.
- 2) S'il a déjà été répondu à la demande d'abandon et si le DSA a gardé les informations permettant de le savoir, une erreur **AbandonError** avec une valeur **tooLate** peut être retournée au demandeur.
- 3) Si la demande d'abandon n'est pas valide, c'est-à-dire si elle vise à abandonner une demande qui n'est pas une demande d'interrogation, il faut retourner au demandeur un message d'erreur **AbandonError** avec une valeur **cannotAbandon** (abandon impossible).
- 4) Si un DSA a des (sous)-demandes chaînées en suspens lorsqu'il reçoit une demande d'abandon valide pour la demande originale, et s'il décide d'essayer d'abandonner, il peut envoyer des demandes d'abandon pour certaines ou toutes les (sous)-demandes en suspens pour l'opération en question (ou pour aucune) puis attendre les réponses à la demande d'abandon et aux (sous)-demandes en suspens. A tout moment au cours de cette opération, le DSA peut envoyer un résultat d'abandon et une erreur **AbandonError** au demandeur puis supprimer les réponses aux demandes d'abandon émises et aux (sous)-demandes en suspens au fur et à mesure qu'elles arrivent.

Si le DSA décide de ne pas envoyer de réponses au demandeur jusqu'à ce qu'il n'y ait plus de (sous)-demandes en suspens, il peut, à titre facultatif, envoyer une erreur **AbandonedFailed** au demandeur s'il a été répondu à toutes les demandes d'abandon émises avec des erreurs **AbandonedFailed** et si aucune opération d'abandon locale n'a été effectuée.

Si une erreur **AbandonedFailed** est retournée au demandeur, la demande originale doit être traitée comme si la demande d'abandon n'avait jamais été reçue.

21 Procédure de fusionnement des résultats

La procédure de fusionnement Result Merging présentée à la Figure 29 est appelée à la suite d'une des procédures Continuation Reference. Cette procédure supprime les duplications si le résultat n'est pas signé; s'il y a des références de continuation supplémentaires dans l'élément **partialOutcomeQualifier.unexplored**, la ou les procédure(s) Continuation Reference pertinente(s) est (sont) appelée(s) si la politique d'exploitation locale le permet.

- 1) Si l'opération est une opération de listage, passer à l'étape 2); si l'opération est une opération de recherche, passer à l'étape 3); sinon, retourner le résultat qui a été fourni comme paramètre d'entrée dans la procédure de fusionnement des résultats.
- 2) L'opération est une opération de listage. Supprimer toutes les duplications.

Si le résultat de l'opération a été émis au niveau local et s'il contient des références de continuation, celles-ci ne seront pas utilisées pour le chaînage mais fournies à l'utilisateur. En pareil cas, passer à l'étape 6).

Si le résultat de l'opération a été reçu à l'issue d'une opération de listage chaînée, il peut contenir des références de continuation. En pareil cas, vérifier si la commande de service **preferChaining** a été sélectionnée. Si elle est mise à **TRUE**, les références de continuation devraient être utilisées pour le chaînage par le DSA. Passer à l'étape 4).

- 3) L'opération est une opération de recherche. Supprimer toutes les duplications. S'il y a un problème de limite, retourner le résultat. Sinon, revenir à l'étape 4).

9594-4 : 1995 (F)

- 4) Traiter chaque référence de continuation figurant dans l'élément **partialOutcomeQualifier.unexplored** du résultat de chaque opération chaînée. Si la politique locale décide de ne pas l'utiliser pour le chaînage, ne pas tenir compte de cette référence de continuation et en choisir une autre. Si la politique locale permet de l'utiliser pour le chaînage, procéder comme suit:

Vérifier la phase de résolution **nameResolutionPhase** fournie dans la référence de continuation. Si elle est **notStarted** ou **proceeding**, l'ajouter à la liste des références de continuation qui sera fournie à la procédure de continuation de résolution du nom (**NRcontinuationList**). Si la phase **NameResolutionPhase** est achevée, ajouter la référence de continuation à la liste des références de continuation qui est fournie à la procédure de Continuation de sous-demande (**SRcontinuationList**).

Continuer jusqu'à ce que toutes les références de continuation aient été traitées.

- 5) Si des références de continuation doivent être traitées dans **SRcontinuationList**, vérifier l'opération. Si l'opération est une opération de listage, appeler la procédure de référence de continuation de listage et revenir à l'étape 2). Si l'opération est une opération de recherche, appeler la procédure de référence de continuation de recherche et revenir à l'étape 3).

Si **SRcontinuationList** est à empty, vérifier s'il y a des références de continuation dans **NRcontinuationList**. Dans l'affirmative, appeler la procédure de référence de continuation de résolution du nom et passer à l'étape 3).

Si les deux listes de continuation sont à empty, passer à l'étape 6).

- 6) Vérifier si le résultat a la valeur empty. Si ce n'est pas le cas, le retourner. Si c'est le cas, retourner un résultat null si la commande d'accès et la politique locale le permettent, ou retourner une erreur appropriée.

22 Procédures d'authentification répartie

Cet article spécifie les procédures nécessaires pour assurer les services d'authentification répartie de l'annuaire. Ces services, et donc ces procédures, relèvent des catégories suivantes:

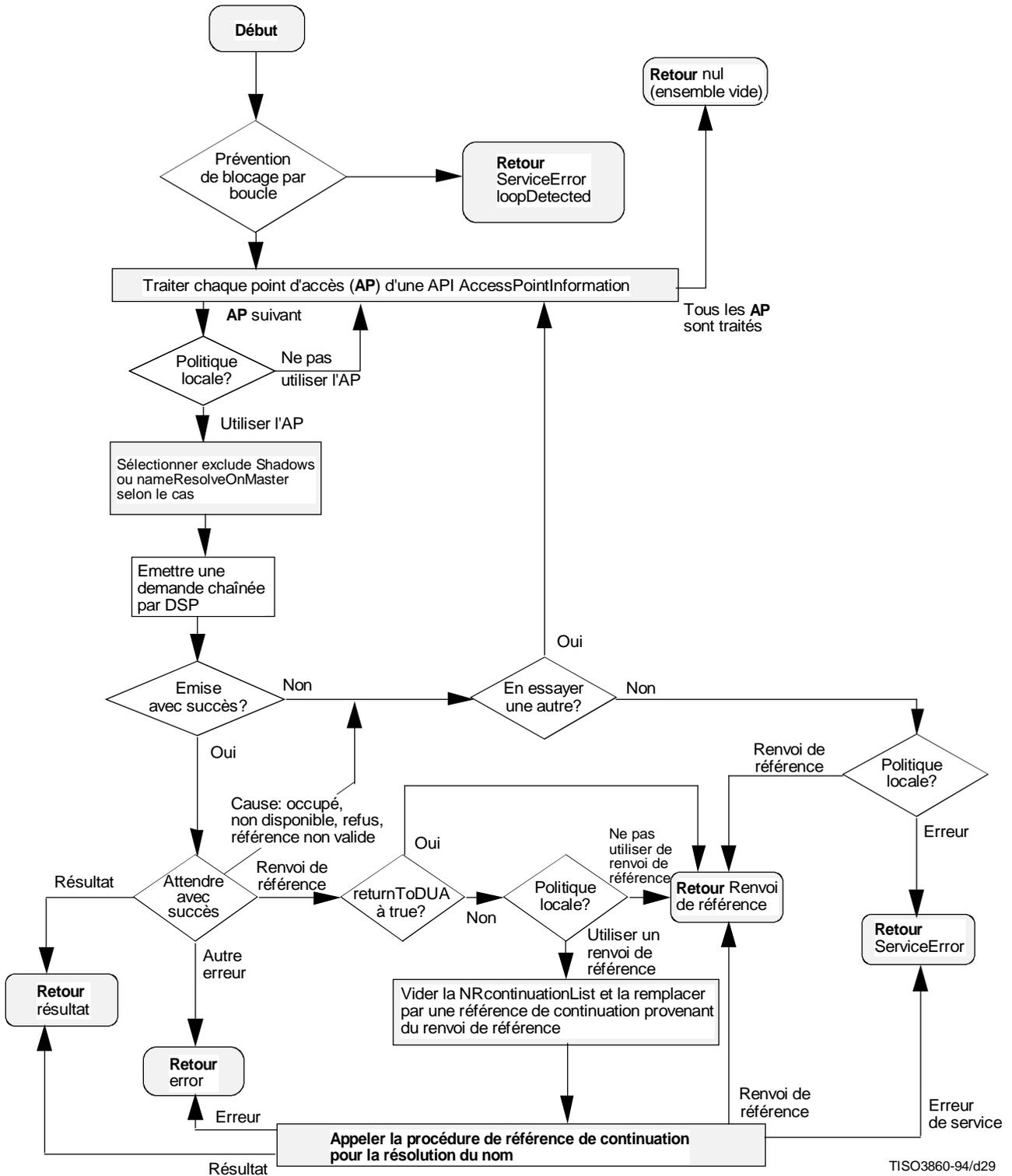
- authentification de l'expéditeur qui est assurée d'une manière non protégée (fondée sur la simple identité) ou sûre (fondée sur des signatures numériques); et
- authentification des résultats, qui est protégée de la même façon (fondée également sur des signatures numériques).

22.1 Authentification de l'expéditeur

22.1.1 Authentification fondée sur l'identité

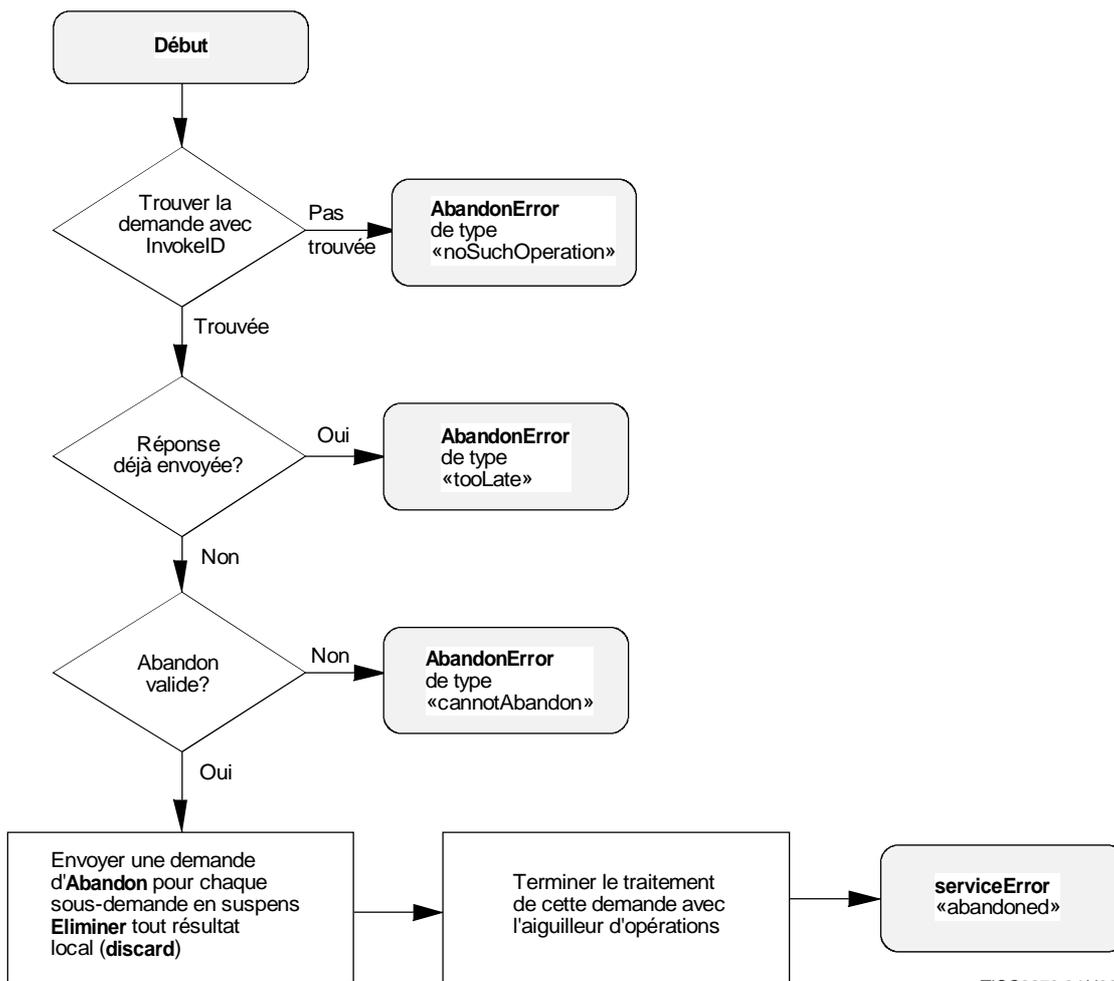
Le service d'authentification fondée sur l'identité permet aux DSA d'authentifier le demandeur initial d'informations en vue d'effectuer des contrôles d'accès locaux. Les DSA qui désirent exploiter ce service doivent adopter la procédure suivante:

- dans le cas d'un DSA demandant l'authentification d'une demande de protocole d'accès à l'Annuaire (DAP) le DSA se procure le nom distinctif du demandeur via les procédures Bind au moment de l'établissement d'une association par le DUA (DUA à DSA). Le succès de ces procédures ne préjuge en aucune façon le niveau d'authentification qui pourra être requis par la suite pour traiter des opérations en utilisant cette association;
- le DSA avec lequel le DUA, établit l'association doit insérer le nom distinctif du demandeur dans le champ initiateur de **ChainingArgument** pour toutes les futures opérations chaînées vers d'autres DSA;
- un DSA, à la réception d'une opération chaînée, peut satisfaire cette opération ou non, selon les droits d'accès établis (par un mécanisme défini au niveau local). Si le résultat n'est pas satisfaisant, une erreur **SecurityError** peut être renvoyée avec la cause **SecurityProblem** mise sur **insufficientAccessRights**.



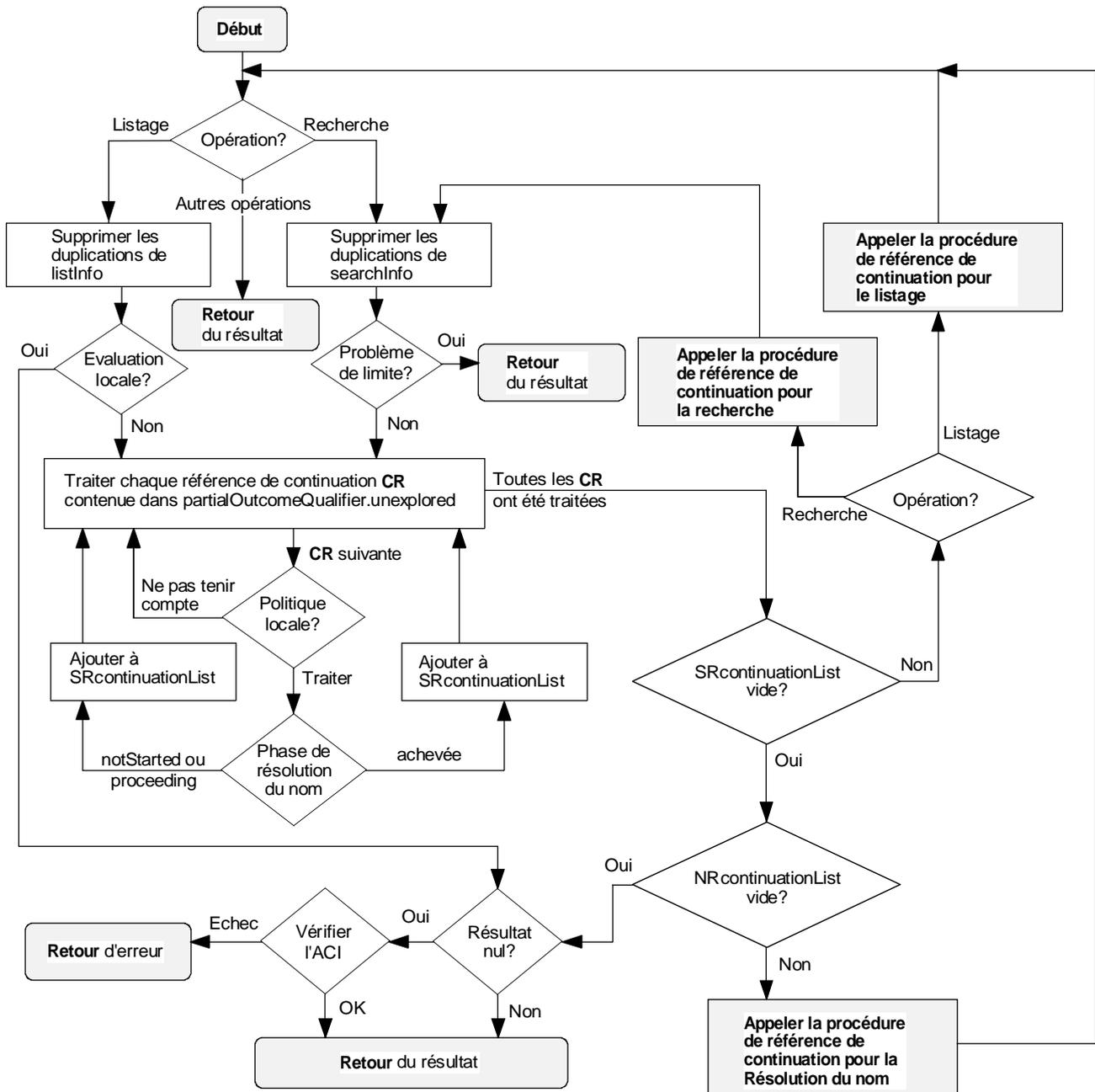
TISO3860-94/d29

Figure 27 – Procédure APInfo



TISO3870-94/d30

Figure 28 – Procédure d'abandon



TISO3880-94/d31

Figure 29 – Procédure de fusionnement des résultats

22.1.2 Authentification de l'expéditeur fondée sur la signature

Le service d'authentification de l'expéditeur fondée sur la signature permet à un DSA d'authentifier (de manière sûre) l'expéditeur d'une demande de service particulière. Les procédures à exécuter par un DSA pour réaliser ce service sont décrites dans le présent article.

Le service d'authentification fondée sur la signature est invoqué par un DUA utilisant la variante **SIGNED** d'une demande de service avec option de signature.

A la réception d'une demande signée de la part d'un autre DSA, un DSA doit enlever cette signature avant de traiter l'opération. En supposant que le résultat de toute vérification de signature s'avérera satisfaisant, le DSA continue à traiter

9594-4 : 1995 (F)

l'opération. Si, durant ce traitement, le DSA a besoin d'effectuer un chaînage, l'ensemble d'arguments de chaque opération chaînée associée sera construit comme suit:

- le DSA forme un ensemble d'arguments qui peut être signé (à titre facultatif); l'ensemble d'arguments comprend l'ensemble des arguments signé entrants ainsi qu'un **ChainingArguments** modifié.

Si le DSA peut fournir des informations pour la réponse, l'authentification de l'expéditeur, fondée sur la demande de service signée, peut être utilisée pour la détermination des droits d'accès à ces informations.

Si un DSA reçoit une demande de service non signée pour des informations dont la diffusion est soumise à l'authentification de l'expéditeur, une **SecurityError** sera retournée avec une cause **SecurityProblem** mise à **protectionRequired**.

22.2 Authentification des résultats

Ce service est fourni pour permettre aux demandeurs d'opérations d'Annuaire (des DUA ou des DSA) de vérifier (de manière sûre, en utilisant des techniques de signature numériques) l'origine des résultats. Le service d'authentification des résultats peut être demandé, que le service d'authentification de l'expéditeur soit utilisé ou non.

On lance le service d'authentification des résultats en utilisant la valeur signée du composant **protectionRequest** contenu dans l'ensemble d'arguments des opérations d'annuaire; un DSA recevant une opération pour laquelle cette option a été choisie pourra alors signer (à titre facultatif) tous les résultats subséquents. L'option signée de la demande de protection sert à indiquer au DSA la préférence des demandeurs; le DSA peut signer ou non un quelconque résultat subséquent.

Si un DSA effectue un chaînage, il dispose d'un certain nombre d'options en ce qui concerne la forme des résultats à renvoyer au demandeur, à savoir:

- a) renvoyer une réponse composite (signée ou non signée) au demandeur;
- b) renvoyer au demandeur un ensemble de deux ou plus de deux réponses partielles non collationnées (signées ou non signées), dans cet ensemble, zéro ou plusieurs éléments peuvent être signés et zéro ou un seul élément peut être non signé. Si un résultat partiel non signé est présent, cet élément peut être en fait une collation d'une ou de plusieurs réponses partielles non signées, provenant d'autres DSA, fournies par ce DSA ou provenant de ces deux sources.

SECTION 6 – ADMINISTRATION DES CONNAISSANCES**23 Vue d'ensemble de l'administration des connaissances**

Pour exploiter un Annuaire largement réparti avec un niveau acceptable de cohérence et de performance, des procédures sont nécessaires pour créer, tenir à jour et étendre les connaissances détenues par chaque DSA. Les mécanismes suivants sont utilisés pour administrer les connaissances d'un DSA.

- a) *Rattachements opérationnels hiérarchiques (HOB) et hiérarchiques non spécifiques (NHOB)* – Ces procédures et protocoles sont définis dans les articles 24 et 25. Ils servent à créer et à tenir à jour des références subordonnées, des références subordonnées non spécifiques, des références supérieures immédiates, ainsi que l'information du préfixe de contexte pour les contextes de dénomination. Ces rattachements opérationnels sont établis entre des DSA maîtres détenant des contextes de dénomination hiérarchiquement reliés les uns aux autres de subordonné immédiat à supérieur immédiat. Le déclenchement des procédures peut être un effet secondaire de la modification du RDN, de l'adjonction ou de la suppression d'une entrée dont le supérieur immédiat n'est pas détenu dans le même DSA que celui qui détient l'entrée.
- b) *Rattachements opérationnels de duplication miroir* – Ces procédures et protocoles sont définis dans la Rec. UIT-T X.525 | ISO/CEI 9594-9. Ils sont utilisés pour créer et tenir à jour les références de connaissance de deux façons. Premièrement, l'effet secondaire de l'établissement (ou de la terminaison) d'accords de duplication miroir se traduit par l'adjonction (ou la suppression) de points d'accès dans les attributs opérationnels **consumerKnowledge** et, à titre facultatif, **secondaryShadow**. Cette information peut ensuite être utilisée par les procédures et les protocoles susmentionnés pour mettre à jour la référence subordonnée dans le DSA maître supérieur et la référence supérieure immédiate dans le DSA maître subordonné. Deuxièmement, le DISP propage les références de connaissance détenues par les DSA maîtres aux DSA consommateurs d'informations miroirs.

- c) *Références croisées* – La distribution des références croisées est une fonction du DSP. La façon dont elle est utilisée pour créer et tenir à jour les références croisées est résumée au 23.2.

NOTE – Les mécanismes utilisés pour établir et tenir à jour la référence supérieure et **myAccessPoint** n'entrent pas dans le cadre de la présente Spécification d'Annuaire.

23.1 Administration des références de connaissances

Le 23.1 décrit comment le DOP est utilisé pour tenir à jour les attributs opérationnels de DSA qui expriment les connaissances. Un exemple simple illustrant la relation entre les attributs de connaissances et les protocoles utilisés pour les tenir à jour est décrit à l'Annexe E.

23.1.1 Administration des connaissances de consommateur par les DSA fournisseurs et maîtres

Une référence de consommateur est exprimée par une valeur de l'attribut **consumerKnowledge**, que détient un DSA fournisseur d'informations miroirs et que l'on associe au préfixe de contexte pour un contexte de dénomination; une référence de fournisseur est exprimée par une valeur de l'attribut **supplierKnowledge**, que détient un DSA consommateur d'informations miroirs et que l'on associe au préfixe de contexte pour un contexte de dénomination. Les deux attributs sont détenus dans des DSE de type **cp**. Une valeur de chacun de ces attributs est créée à l'établissement du rattachement opérationnel de duplication miroir et mise à jour au moment de la modification de ce rattachement.

Un DSA fournisseur peut obtenir les informations nécessaires pour construire les valeurs de l'attribut **secondaryShadows** si le composant facultatif **secondaryShadows** de son accord **ShadowingAgreement** avec un consommateur a la valeur TRUE. Dans ce cas, chaque fois que le DSA du consommateur détectera une modification (par addition, modification ou suppression de points d'accès) de l'ensemble des DSA détenant des copies de la zone dupliquée d'usage commun (ses consommateurs ou, tout à tour, les consommateurs de ses consommateurs, etc. jusqu'au niveau le plus bas auquel des duplications miroirs peuvent être effectuées), cet agent communiquera ces nouvelles informations (sous forme d'ensemble de fournisseurs et de consommateurs **SuppliersAndConsumers**) au moyen d'une opération de type **modifyOperationalBinding** comme décrit dans la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Un DSA fournisseur tient à jour son propre attribut **secondaryShadows** associé au préfixe de contexte de la manière suivante:

- a) L'ensemble de fournisseurs et de consommateurs (**SuppliersAndConsumers**) reçu d'un consommateur au moyen d'une opération de type **modifyOperationalBinding** peut servir à créer ou à remplacer des valeurs de l'attribut **secondaryShadows**. Le composant **supplier** de l'ensemble **SuppliersAndConsumers** représente le point d'accès d'un DSA de consommateur (ou de ses consommateurs, etc., selon le niveau le plus bas des duplications miroirs secondaires); le composant **consumers** de l'ensemble **SuppliersAndConsumers** représente l'ensemble des consommateurs du consommateur (ou de leurs consommateurs, etc., selon le niveau le plus bas des duplications miroirs secondaires).
- b) Chaque consommateur qui communique à son fournisseur une opération de type **modifyOperationalBinding** contenant un ensemble du type **SuppliersAndConsumers** présentera les valeurs de son attribut **secondaryShadows** et une valeur nouvellement construite au moyen de son propre point d'accès: **myAccessPoint** (en tant que composant **supplier**) et au moyen des valeurs des points d'accès des consommateurs, contenues dans l'attribut **consumerKnowledge**, qui représentent les consommateurs détenant (en tant que composant **consumer**) des duplications miroirs d'usage commun.

L'itération de cette procédure permet à un DSA maître, pour un contexte de dénomination, de connaître tous ses DSA consommateurs d'informations miroirs secondaires détenant des zones dupliquées d'usage commun obtenues à partir du contexte de dénomination. Cette information est ensuite disponible pour l'administration des références subordonnées, subordonnées non spécifiques et supérieures immédiates.

23.1.2 Administration des connaissances subordonnées et supérieures immédiates dans les DSA maîtres

Une référence subordonnée est exprimée par une valeur de l'attribut **specificKnowledge**, détenue dans une DSE de type **subr** par le DSA qui détient le contexte de dénomination immédiatement supérieur par rapport à celui qui est référencé; une référence supérieure immédiate est exprimée par une valeur de l'attribut **specificKnowledge**, que détient dans une DSE de type **immSupr** le DSA qui détient le contexte de dénomination immédiatement subordonné à celui qui est référencé. Une valeur de chacun de ces attributs est déterminée dans les DSA maîtres supérieur et subordonné à l'établissement du HOB, et elle est actualisée au moment de la modification de ce rattachement.

9594-4 : 1995 (F)

Un DSA maître subordonné fournit à un DSA maître supérieur l'information requise pour déterminer la référence subordonnée par l'intermédiaire du composant **accessPoints** du paramètre **SubordinateToSuperior** qu'il transmet au supérieur dans le DOP. L'information figurant dans **accessPoints** est déterminée par les valeurs des attributs que détient le DSA subordonné, de la manière suivante:

- a) La valeur de l'attribut **myAccessPoint** (que détient la DSE racine) est utilisée pour constituer l'élément dans **accessPoints** avec la valeur **master** pour **category**.
- b) Les valeurs de **consumerKnowledge** et **secondaryShadows** (que détient la DSE du préfixe de contexte subordonné) sont utilisées pour former des éléments supplémentaires dans **accessPoints**, avec la valeur **shadow** pour **category**.

Un DSA maître supérieur fournit à un DSA maître subordonné l'information requise pour construire sa référence supérieure immédiate à l'aide du composant **contextPrefixInfo** du paramètre **SuperiorToSubordinate** qu'il transfère au subordonné dans le DOP. Ce composant est une valeur du type **SEQUENCE OF Vertex**, qui contient une séquence d'éléments correspondant au trajet depuis la racine du DIT jusqu'au préfixe de contexte subordonné. Pour l'un de ces éléments, correspondant au préfixe du contexte de dénomination immédiatement supérieur, le composant facultatif **accessPoints** est présent. Le DSA subordonné détient cette information sous la forme d'un attribut **specificKnowledge** dans la DSE, de type **immSupr**, correspondant à cet élément de **contextPrefixInfo**. L'information incluse dans **accessPoints** par le DSA supérieur est déterminée par les valeurs des attributs que détient le DSA supérieur, de la manière suivante:

- a) La valeur de l'attribut **myAccessPoint** (que détient la DSE racine) est utilisée pour constituer l'élément dans **accessPoints** avec la valeur **master** pour **category**.
- b) Les valeurs de **consumerKnowledge** et **secondaryShadows** (que détient la DSE du préfixe de contexte subordonné) sont utilisées pour former des éléments supplémentaires dans **accessPoints**, avec la valeur **shadow** pour **category**.

NOTE – Seuls les points d'accès correspondant aux DSA consommateurs recevant des zones dupliquées d'usage commun doivent être choisis par les DSA supérieurs et subordonné à partir de leurs attributs **consumerKnowledge** pour être inclus dans **accessPoints**. Les procédures de détermination de **secondaryShadows** garantissent que ces points d'accès identifieront les DSA de duplication miroir détenant des zones dupliquées d'usage commun.

23.1.3 Administration des connaissances subordonnées et supérieures immédiates dans les DSA consommateurs

Un DSA consommateur d'informations miroirs concluant un accord avec son fournisseur pour recevoir les connaissances supérieures immédiates et subordonnées associées avec une unité de duplication, s'engage de telle sorte que ses références supérieures immédiates et subordonnées soient tenues à jour par son DSA fournisseur d'informations miroirs via le DISP.

NOTE – Pour certaines spécifications d'unités de duplication, il peut être nécessaire que le DSA consommateur conclue un accord pour recevoir **extendedKnowledge** afin que des connaissances subordonnées puissent lui être fournies par son fournisseur.

23.2 Demande de références croisées

Pour améliorer les performances du système d'Annuaire, l'ensemble local de références croisées peut être étendu par l'utilisation des opérations d'Annuaire ordinaires. Si un DSA supporte le DSP, il peut demander à un autre DSA (qui doit également supporter le DSP) de retourner les références de connaissance contenant des informations sur l'emplacement des contextes de dénomination relatifs au nom d'objet cible d'une opération d'Annuaire ordinaire.

Si le composant **returnCrossRefs** de **ChainedOperationsArgument** est mis à **TRUE**, le composant **CrossReference** de **ChainedOperationsResult** peut être présent; il comprend une séquence d'éléments de références croisées.

Si un DSA ne peut pas chaîner une demande au DSA suivant, un renvoi de référence est retourné au DSA d'origine. Si le composant **returnCrossRefs** de **ChainingArguments** est mis à **TRUE**, le renvoi de référence peut contenir en outre le préfixe du contexte de dénomination auquel se réfère le renvoi de référence. Le composant **contextPrefix** est absent si le renvoi de référence est fondé sur une référence subordonnée non spécifique. La référence croisée retournée par un renvoi de référence est fondée sur les connaissances détenues par le DSA qui émet le renvoi de référence.

Dans les deux cas (résultat du chaînage et renvoi de référence), une autorité administrative peut, par le biais de son DSA, choisir d'ignorer la demande de retour de références croisées.

23.3 Incohérences de connaissances

L'Annuaire doit disposer de mécanismes de contrôle de cohérence, afin de garantir un certain niveau de cohérence des connaissances.

NOTE – Dans certaines circonstances, les références de connaissances sont précises (il ne s'agit pas de références non valides telles qu'elles sont décrites ci-après), mais elles ne sont pas valides pour être utilisées par un DSA parce que le DMD du DSA auquel il est fait référence ne souhaite pas du tout que le DSA référençant entre en contact avec lui (par exemple DSA ayant acquis d'une manière ou d'une autre une référence croisée au DSA référencé) ou ne souhaite pas que l'on entre en relation avec lui pour jouer un rôle particulier (par exemple DSA maître pour un contexte de dénomination).

23.3.1 Détection des incohérences de connaissances

Les incohérences et leur détection dépendent des types de références de connaissances:

- a) *Références croisées et subordonnées* – Ce type de référence n'est pas valide si le DSA référencé ne détient pas de contexte de dénomination ou de zone reproduite obtenue à partir du contexte de dénomination dont le préfixe de contexte est contenu dans la référence. Cette incohérence sera détectée durant le processus de résolution du nom, par inspection des composants **operationProgress** et **referenceType** de **ChainingArgument**.
- b) *Références subordonnées non spécifiques* – Ce type de référence n'est pas valide si le DSA référencé ne détient pas de contexte de dénomination local dont le préfixe de contexte est contenu dans la référence moins le dernier RDN. Le contrôle de cohérence est appliqué comme indiqué ci-dessus.
- c) *Références supérieures* – Une référence supérieure non valide est une référence qui ne fait pas partie d'un chemin de référence vers la racine. La tenue à jour des références supérieures doit être assurée par des moyens extérieurs et n'entre pas dans le cadre de la présente Spécification d'Annuaire.

NOTE – Il n'est pas toujours possible de détecter une référence supérieure non valide.

- d) *Références supérieures immédiates* – Ce type de référence n'est pas valide si le DSA référencé ne détient pas un contexte de dénomination ou une zone reproduite que l'on obtient à partir du contexte de dénomination avec le préfixe de contexte contenu dans la référence. En outre, l'utilisation d'un tel type de référence est uniquement valide lorsque le composant **operationProgress** de **ChainingArguments** a la valeur **notStarted** ou **proceeding**. Cette incohérence sera détectée pendant le processus de résolution du nom par inspection des composants **operationProgress** et **referenceType** de **ChainingArguments**.
- e) *Références du fournisseur* – Ce type de référence, qui identifie le fournisseur d'une zone reproduite et, facultativement, le maître pour le contexte de dénomination à partir duquel est obtenue la zone reproduite, n'est pas valide si le DSA référencé n'est pas le fournisseur d'informations miroirs pour le DSA utilisant la référence (lorsque le composant **referenceType** de **ChainingArguments** a la valeur **supplier**), ou si le DSA référencé n'est pas le maître pour le contexte de dénomination (lorsque **referenceType** a la valeur **master**). Cette incohérence sera détectée pendant les phases de résolution du nom et d'évaluation du traitement des opérations par inspection du composant **referenceType** de **ChainingArguments**.

23.3.2 Signalisation des incohérences de connaissances

Si un chaînage est utilisé pour traiter une demande présentée à l'Annuaire, toutes les incohérences de connaissances seront détectées par le DSA qui détient la référence de connaissance non valide, par la réception d'une **serviceError**, avec la cause **invalidReference**.

Si un DSA retourne un renvoi de référence fondé sur une référence de connaissance non valide, une **serviceError** sera retournée au demandeur avec la cause **invalidReference**, s'il utilise le renvoi de référence. La façon dont l'état d'erreur sera communiquée au DSA qui enregistre la référence non valide n'entre pas dans le cadre de la présente Spécification d'Annuaire.

23.3.3 Traitement des références de connaissances incohérentes

Quand un DSA a détecté une référence non valide, il doit essayer de rétablir la cohérence des connaissances. Pour cela, il peut par exemple supprimer simplement une référence croisée non valide, ou la remplacer par une référence croisée correcte, qui peut être obtenue à l'aide des mécanismes **returnCrossRefs**.

La façon dont un DSA traite effectivement les références non valides relève d'une initiative locale et n'entre pas dans le cadre de la présente Spécification d'Annuaire.

9594-4 : 1995 (F)

24 Rattachements opérationnels hiérarchiques

Un rattachement opérationnel hiérarchique (HOB) est utilisé pour représenter la relation entre deux DSA détenant deux contextes de dénomination, l'un étant immédiatement subordonné à l'autre. S'agissant d'un HOB, le DSA supérieur détient une référence subordonnée au contexte de dénomination que possède le DSA subordonné; le DSA subordonné détient une référence supérieure immédiate au contexte de dénomination que possède le DSA supérieur. Le rattachement opérationnel garantit que les informations de connaissances appropriées seront échangées et tenues à jour entre les deux DSA de sorte qu'ils puissent agir pendant le processus de résolution du nom et d'évaluation de l'opération, comme défini dans les articles 18 et 19.

24.1 Caractéristiques du type de rattachement opérationnel

24.1.1 Symétrie et rôles

Le rattachement opérationnel hiérarchique est de type asymétrique. Les deux rôles d'un rattachement de ce type sont les suivants:

- a) le rôle du DSA maître pour le contexte de dénomination supérieur, *le DSA supérieur* (associé au rôle abstrait «A»); et
- b) le rôle du DSA maître pour le contexte de dénomination subordonné, *le DSA subordonné* (associé au rôle abstrait «B»).

24.1.2 Accord

L'information d'accord échangée pendant l'établissement du rattachement opérationnel hiérarchique est une valeur de **HierarchicalAgreement** qui contient le nom distinctif relatif du nouveau préfixe de contexte (le composant **rdn**) et le nom distinctif de l'entrée immédiatement supérieure au nouveau contexte de dénomination (le composant **immediateSuperior**). Cette information est transmise par le DSA qui établit le HOB.

```
HierarchicalAgreement ::= SEQUENCE {
    rdn [0] RelativeDistinguishedName,
    immediateSuperior [1] DistinguishedName }
```

24.1.3 Initiateur

24.1.3.1 Etablissement

L'établissement d'un rattachement opérationnel hiérarchique peut être entrepris par l'un ou l'autre DSA. L'établissement par le DSA supérieur peut résulter d'une opération adjonction d'entrée avec la spécification du DSA subordonné dans l'extension **targetSystem**, ou bien d'une intervention administrative. L'établissement par le DSA subordonné (qui connecte au DIT global une entrée ou une sous-arborescence existant au niveau local) résulte d'une intervention administrative.

24.1.3.2 Modification

La modification d'un rattachement opérationnel hiérarchique peut être entreprise par l'un ou l'autre DSA. Le DSA supérieur peut apporter cette modification à la suite d'une modification de l'information du préfixe de contexte supérieur. Cela peut résulter de l'une quelconque des opérations de modification, ou d'une intervention administrative.

Chaque DSA peut modifier l'accord suite à une modification du RDN de l'entrée du préfixe de contexte du contexte de dénomination subordonné. Le DSA supérieur procède à cette modification en raison d'un changement de nom distinctif relatif plus haut dans le DIT ou à cause d'une intervention administrative. Le DSA subordonné procède à la modification de l'accord à cause d'une opération ModifyDN (modifier nom distinctif) effectuée sur un préfixe de contexte ou suite à une intervention administrative.

Chaque DSA peut également modifier le HOB si l'information de point d'accès pour son contexte de dénomination change.

24.1.3.3 Terminaison

La terminaison d'un rattachement opérationnel hiérarchique peut être effectuée par l'un ou l'autre DSA. La terminaison par le DSA supérieur peut résulter d'une intervention administrative. La terminaison par le DSA subordonné peut résulter d'une opération de suppression d'entrée supprimant l'entrée du préfixe de contexte du contexte de dénomination subordonné, ou bien d'une intervention administrative.

24.1.4 Paramètres d'établissement

Les paramètres d'établissement pour les deux rôles d'un HOB (DSA supérieur et DSA subordonné) sont différents. Le paramètre d'établissement pour le rôle de DSA supérieur est une valeur de **SuperiorToSubordinate**; pour le DSA subordonné, le paramètre d'établissement est une valeur de **SubordinateToSuperior**.

24.1.4.1 Paramètre d'établissement de DSA supérieur

Le paramètre d'établissement émis par le DSA supérieur, qui est une valeur de **SuperiorToSubordinate**, fournit au DSA subordonné des informations concernant les nœuds du DIT supérieurs au préfixe de contexte du nouveau contexte de dénomination (qui comprend la référence supérieure immédiate) et, facultativement, des attributs d'utilisateur et opérationnels pour l'entrée du préfixe de contexte subordonné ainsi que des copies d'attributs d'utilisateur et opérationnels provenant de l'entrée immédiatement supérieure au nouveau préfixe de contexte.

```

SuperiorToSubordinate ::= SEQUENCE {
    contextPrefixInfo[0]    DITcontext,
    entryInfo             [1]    SET OF Attribute OPTIONAL,
    immediateSuperiorInfo [2]    SET OF Attribute OPTIONAL }
    
```

24.1.4.1.1 Information du préfixe de contexte

Le composant **contextPrefixInfo** de **SuperiorToSubordinate** est une valeur du type **DITcontext**, qui correspond à une séquence de valeurs **Vertex**.

```

DITcontext ::= SEQUENCE OF Vertex

Vertex ::= SEQUENCE {
    rdn         [0]    RelativeDistinguishedName,
    admPointInfo [1]    SET OF Attribute OPTIONAL,
    subentries  [2]    SET OF SubentryInfo OPTIONAL,
    accessPoints [3]    MasterAndShadowAccessPoints OPTIONAL }
    
```

Le composant **contextPrefixInfo** est essentiellement la séquence de RDN qui constitue le nom distinctif du nouveau préfixe de contexte, chaque RDN (donné par le composant **rdn**) étant facultativement accompagné par des informations supplémentaires.

Le composant facultatif **admPointInfo** d'un **Vertex** indique que le nœud du DIT est un point administratif et fournit au moins son attribut opérationnel **administrative-role**.

L'information de sous-entrée associée à un point administratif est fournie par le composant **subentries** d'un nœud **Vertex**, à savoir une série d'une ou plusieurs valeurs **SubentryInfo**. Chaque valeur **SubentryInfo** est composée du RDN de la sous-entrée (composant **rdn**) et des attributs de la sous-entrée (composant **info**).

```

SubentryInfo ::= SEQUENCE {
    rdn         [0]    RelativeDistinguishedName,
    info        [1]    SET OF Attribute }
    
```

Le composant facultatif **accessPoints** d'un **Vertex** indique que ce nœud correspond au préfixe de contexte du contexte de dénomination immédiatement supérieur. Le supérieur utilise ce composant pour fournir au subordonné l'information requise pour sa référence supérieure immédiate.

24.1.4.1.2 Information d'entrée

Le composant facultatif **entryInfo** de **SuperiorToSubordinate** est une série d'attributs établissant le contenu de la nouvelle entrée du préfixe de contexte.

24.1.4.1.3 Information d'entrée supérieure immédiate

Le composant facultatif **immediateSuperiorInfo** de **SuperiorToSubordinate** est une copie d'une série d'attributs, en particulier **objectClass** et **entryACI**, provenant de l'entrée immédiatement supérieure au nouveau préfixe de contexte.

NOTE – Ce composant peut être utilisé par le subordonné pour optimiser l'évaluation d'une demande de listage qui produit un résultat **ListResult** de valeur empty pour une entrée d'objet de base qui est l'entrée immédiatement supérieure au préfixe de contexte subordonné [voir la Note du 19.3.1.2.2, alinéa 2)].

9594-4 : 1995 (F)

24.1.4.2 Paramètre d'établissement du DSA subordonné

Le paramètre d'établissement émis par le DSA subordonné, qui est une valeur de **SubordinateToSuperior**, fournit au DSA supérieur des informations concernant le contexte de dénomination subordonné.

```
SubordinateToSuperior ::= SEQUENCE {
    accessPoints [0] MasterAndShadowAccessPoints OPTIONAL,
    alias [1] BOOLEAN DEFAULT FALSE,
    entryInfo [2] SET OF Attribute OPTIONAL }
```

Le composant **accessPoints** de **SubordinateToSuperior** est utilisé par le subordonné en vue de fournir au supérieur les informations requises pour sa référence subordonnée.

Le composant **alias** de **SubordinateToSuperior** est utilisé pour indiquer au supérieur que le contexte de dénomination subordonné comprend une entrée d'alias unique.

Le composant **entryInfo** de **SubordinateToSuperior** comprend une copie d'une série d'attributs, en particulier **objectClass** et **entryACI**, provenant de la nouvelle entrée du préfixe de contexte.

NOTE – Ces deux derniers composants peuvent être utilisés par le supérieur pour optimiser l'évaluation d'une demande de listage dont l'entrée d'objet de base est l'entrée immédiatement supérieure au préfixe de contexte subordonné.

24.1.5 Paramètres de modification

Pour les modifications d'un HOB, le paramètre de modification de DSA supérieur, **SuperiorToSubordinateModification**, est **SuperiorToSubordinate**, à une restriction près qui est l'absence éventuelle du composant **entryInfo**; le paramètre de modification du rôle de DSA subordonné est **SubordinateToSuperior**.

```
SuperiorToSubordinateModification ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT})
```

Ces paramètres sont identiques (moyennant la restriction susmentionnée) aux paramètres d'établissement correspondants qui sont utilisés pour indiquer les modifications apportées aux informations fournies dans les paramètres d'établissement suite à l'établissement d'un HOB.

Si un composant quelconque de **SuperiorToSubordinate** (ou, ultérieurement, **SuperiorToSubordinateModification**) ou de **SubordinateToSuperior** fait l'objet d'une modification (par exemple composant **contextPrefixInfo** de **SuperiorToSubordinate**), le composant correspondant du paramètre de modification (par exemple composant **contextPrefixInfo** de **SuperiorToSubordinateModification**) doit être fourni dans son intégralité à l'intérieur de la modification du rattachement opérationnel.

24.1.6 Paramètres de terminaison

Pour la terminaison d'un HOB, aucun des deux DSA ne fournit de paramètre de terminaison.

24.1.7 Identification du type

Le rattachement opérationnel hiérarchique est identifié par l'identificateur d'objet assigné lors de la définition de l'objet d'information **hierarchicalOperationalBinding OPERATIONAL-BINDING**, au 24.2.

24.2 Définition de la classe d'objets d'information de type operational-binding

Cet article définit le type de rattachement opérationnel hiérarchique en utilisant le modèle de classe d'objets d'information **OPERATIONAL-BINDING** défini dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.

```
hierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT HierarchicalAgreement
    APPLICATION CONTEXTS {
        {directorySystemAC}}
    ASYMMETRIC
    ROLE-A { -- DSA supérieur
        ESTABLISHMENT-INITIATOR TRUE
        ESTABLISHMENT-PARAMETER SuperiorToSubordinate
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER SuperiorToSubordinateModification
        TERMINATION-INITIATOR TRUE}
```

```

ROLE-B { -- DSA subordonné
    ESTABLISHMENT-INITIATOR TRUE
    ESTABLISHMENT-PARAMETER SubordinateToSuperior
    MODIFICATION-INITIATOR TRUE
    MODIFICATION-PARAMETER SubordinateToSuperior
    TERMINATION-INITIATOR TRUE }
ID hierarchicalOperationalBindingID }
    
```

24.3 Procédures de DSA pour la gestion des rattachements opérationnels hiérarchiques

Dans les procédures suivantes, une nouvelle DSE ou une marque (c'est-à-dire une indication d'état associée à un élément d'information donné) créée par un DSA doit être stockée dans une mémoire stable. Ce faisant, il devient possible pour les deux DSA suivant les procédures ci-après de maintenir une compréhension cohérente des paramètres du HOB en cas de défaillance de la communication et des systèmes/terminaux.

Dans la procédure d'établissement et de modification décrite ci-après, le DSA jouant le rôle de celui qui répond (c'est-à-dire n'entretenant pas l'établissement ou la modification) peut fournir au DSA qui joue le rôle initiateur des informations (par exemple attributs opérationnels) qui ne sont pas acceptables pour telle ou telle raison. En pareil cas, le DSA initiateur peut mettre fin au rattachement opérationnel.

24.3.1 Procédure d'établissement

24.3.1.1 Etablissement entrepris par un DSA supérieur

Si un DSA évalue une opération Add Entry avec un DSA différent, spécifié dans l'extension **targetSystem**, il devra établir un rattachement opérationnel hiérarchique conformément à la procédure suivante. Si un DSA, pour des raisons administratives, désire établir un HOB avec un DSA subordonné et s'il admet le DOP par HOB, la procédure ci-après doit être suivie:

- 1) Le DSA supérieur crée une nouvelle DSE de type **subr** avec le nom de la nouvelle entrée, et marque cette nouvelle DSE comme *étant ajoutée*. Le DSA supérieur émet une identification **bindingID** unique et l'enregistre avec la nouvelle DSE.
- 2) Le DSA supérieur doit envoyer au DSA subordonné une opération établissement de rattachement opérationnel contenant les paramètres suivants:
 - a) type de rattachement **bindingType** mis à **hierarchicalOperationalBindingID**;
 - b) paramètre d'établissement **SuperiorToSubordinate** avec les composants **contextPrefixInfo** et **entryInfo** présents; tous les autres paramètres sont facultatifs;
 - c) **HierarchicalAgreement** avec le composant **immediateSuperior** mis au nom distinctif du supérieur immédiat de la nouvelle entrée et le composant **rdn** mis au RDN de la nouvelle entrée;
 - d) paramètres **bindingID**, **myAccessPoint** et **valid**, le cas échéant.
- 3) Si le DSA subordonné accepte l'opération, il crée les DSE requises de type **glue**, **subentry**, **admPoint**, **rhob** et **immSupr**, selon le cas, pour représenter l'information **contextPrefixInfo**, une DSE de type **cp** et **entry** ou **alias** pour représenter le nouvel objet du préfixe de contexte ou l'entrée d'alias; et, le cas échéant, une DSE de type **rhob** et **entry** pour représenter l'information **immediateSuperiorInfo**. Il enregistre **bindingID** avec la DSE de la nouvelle entrée du préfixe de contexte et retourne un paramètre **SubordinateToSuperior** au DSA supérieur.

Si le DSA subordonné refuse l'opération, il renvoie une erreur de rattachement opérationnel avec la valeur de cause appropriée.

Si le contexte de dénomination existe déjà et si les valeurs de l'identificateur **bindingID** sont les mêmes pour le contexte existant et pour le nouveau contexte, c'est que le DSA subordonné a déjà créé le contexte de dénomination demandé, auquel cas il doit retourner un résultat au DSA supérieur. Si les valeurs ne sont pas les mêmes, une erreur de rattachement opérationnel avec la cause **invalidAgreement** est envoyée; cela signifie que le DSA supérieur a une incohérence de connaissances permanente qui doit être corrigée par un administrateur.

- 4) Si le DSA supérieur reçoit une erreur, il supprime la DSE marquée du type **subr** et retourne une erreur pour l'opération Add Entry.

Si le DSA supérieur reçoit un résultat, il supprime l'indication de la DSE qui représente le **subr** et retourne un résultat pour l'opération Add Entry.

9594-4 : 1995 (F)

En cas de défaillance (par exemple: communication du système terminal), le DSA supérieur doit répéter les opérations à partir de l'étape 2) jusqu'à l'obtention d'un résultat ou d'une erreur pour chaque établissement en suspens d'un rattachement opérationnel hiérarchique dont il est l'initiateur. Si l'établissement résulte d'une opération Add Entry et si le demandeur interrompt l'opération (par exemple en libérant ou en interrompant l'association d'application) avant que l'établissement soit terminé, le DSA supérieur ne doit pas tenir compte de cet événement et doit terminer l'établissement (qui peut réussir ou non). En pareil cas, l'utilisateur ne sera pas informé du résultat de l'opération Add Entry.

NOTES

1 Le marquage de la référence subordonnée aide à gérer les reprises et les annulations. Un autre utilisateur ne peut ajouter une entrée qui est déjà marquée et le DSA répète l'établissement de rattachement opérationnel pour toutes les références subordonnées marquées, après une défaillance.

2 Compte tenu de la procédure décrite ci-dessus, les connaissances n'ont qu'une incohérence transitoire. La façon dont le DSA supérieur traite les opérations indépendantes qui lisent la référence subordonnée lorsqu'elle est marquée, relève d'une initiative locale.

24.3.1.2 Etablissement entrepris par un DSA subordonné

Le DSA subordonné peut entreprendre un rattachement opérationnel hiérarchique. Cela peut faire suite au désir d'un administrateur de relier, à un certain point du DIT global, un sous-arbre d'entrées détenues dans le DSA. En pareil cas, ce DSA subordonné doit établir un HOB conformément à la procédure suivante:

- 1) Le DSA subordonné a une DSE de type **cp** en tant que partie d'un contexte de dénomination existant, ou il en crée une nouvelle. Il marque la DSE *ajoutée*, émet un identificateur **bindingID** unique et l'enregistre avec la DSE du préfixe de contexte.
- 2) Le DSA subordonné envoie une opération d'établissement de rattachement opérationnel au DSA supérieur, contenant les paramètres suivants:
 - a) **bindingType** mis à **hierarchicalOperationalBindingID**;
 - b) paramètre d'établissement **SubordinateToSuperior**, le cas échéant;
 - c) **HierarchicalAgreement** avec le composant **immediateSuperior** mis au nom distinctif du supérieur immédiat de la nouvelle entrée et le composant **rdn** mis au RDN de la nouvelle entrée;
 - d) paramètres **bindingID**, **myAccessPoint** et **valid**, le cas échéant.

Si le DSA supérieur refuse l'opération, il retourne une erreur de type Operational Binding avec la valeur de cause appropriée.
- 3) Le DSA supérieur vérifie qu'il est maître pour le supérieur immédiat de la nouvelle entrée du préfixe de contexte ou retourne une erreur de rattachement opérationnel avec la cause **roleAssignment**.
- 4) Le DSA supérieur vérifie que le RDN demandé pour le nouveau préfixe de contexte n'est pas déjà utilisé. Si l'on ne trouve pas de RDN concordant en utilisant l'information détenue localement, mais que la DSE immédiatement supérieure soit du type **nssr**, la procédure décrite au 15.7 est appliquée. Si l'on ne trouve toujours pas de RDN concordant en utilisant cette procédure, le DSA supérieur crée une DSE de type **subr**, enregistre avec cette entrée **bindingID** et retourne un résultat.

Si l'on trouve une référence subordonnée avec ce RDN, on compare les deux valeurs de **bindingID**. Si ces valeurs sont les mêmes, un résultat est retourné. Le paramètre **SuperiorToSubordinate** retourné par le DSA supérieur ne doit pas contenir le composant **entry**. Si les deux valeurs de **bindingID** ne sont pas identiques, une erreur de rattachement opérationnel est envoyée avec la cause **invalidAgreement**; cela signifie que le DSA supérieur a une incohérence de connaissances permanente qui doit être corrigée par un administrateur.

Si un RDN concordant est trouvé dans l'examen d'une NSSR, une erreur de rattachement opérationnel est envoyée avec la cause **invalidAgreement**; cela signifie également que le DSA supérieur a une incohérence de connaissances permanente qui doit être corrigée par un administrateur.

- 5) Si le DSA subordonné reçoit une erreur, il supprime la nouvelle DSE du préfixe de contexte et sa marque. Le sort de l'information d'entrée à partir de laquelle a été obtenue la DSE du préfixe de contexte doit être déterminé localement.

Si le DSA subordonné reçoit un résultat, il ajoute les DSE nécessaires de type **glue**, **subentry**, **admPoint**, **rhob** et **immSupr**, selon le cas, pour représenter l'information **contextPrefixInfo** et, le cas échéant, une DSE de type **rhob** et **entry** pour représenter l'information **immediateSuperiorInfo**. La marque de la DSE du préfixe de contexte est supprimée.

En cas de défaillance (par exemple communication du système terminal), le DSA subordonné reprend la procédure à partir de l'étape 2) jusqu'à ce qu'il ait reçu un résultat ou une erreur pour toutes les opérations d'établissement en suspens d'un HOB dont il est l'initiateur.

24.3.2 Procédure de modification

Les procédures ci-après sont définies pour la modification d'un HOB entrepris à l'aide de la procédure décrite au 24.3.1.

24.3.2.1 Procédure de modification entreprise par un DSA supérieur

On peut invoquer cette procédure à la suite des opérations de modification, comme décrit au 19.1, ou à la suite d'une intervention administrative (par exemple pour transmettre les modifications relatives aux paramètres **myAccessPoint**, **agreement** ou **valid** du HOB). Par ailleurs, si un DSA supérieur détecte des modifications dans les composants **contextPrefixInfo** ou **immediateSuperiorInfo** de la valeur **SuperiorToSubordinate** qu'il a fournie au DSA subordonné, il doit faire suivre ces nouvelles informations au DSA subordonné en utilisant la procédure ci-après:

- 1) Marquer la DSE de type **subr** comme *étant modifiée* et, si cette modification résulte d'une modification apportée au RDN de l'entrée du préfixe de contexte subordonné, une nouvelle DSE de type **subr** est ajoutée et marquée comme *étant ajoutée*.
- 2) Le DSA supérieur produit une nouvelle valeur de **bindingID** à partir de la valeur existante en augmentant son composant **version**. En utilisant ce nouveau **bindingID**, il envoie au DSA subordonné une opération de modification du rattachement opérationnel avec le paramètre de modification **SuperiorToSubordinateModification**.
- 3) Le DSA subordonné vérifie le composant **identifiant** de **bindingID**. S'il n'a pas d'accord de ce genre avec le DSA supérieur, ou si le composant **version** a une valeur inférieure à celle de la version du HOB, il doit retourner une erreur de rattachement opérationnel avec la cause **InvalidAgreement**.
- 4) Le DSA subordonné peut accepter la modification du HOB, modifier ou reconstruire les DSE représentant les informations du préfixe de contexte, actualiser le composant **version** de son **bindingID** et retourner un résultat. Sinon, il peut retourner une erreur et mettre fin à l'accord.
- 5) Si le DSA supérieur reçoit un résultat, la modification est achevée. Si cette modification résulte d'une modification apportée au RDN de l'entrée du préfixe de contexte subordonné, on supprime la marque de la nouvelle DSE de type **subr** marquée comme *étant ajoutée*, et l'ancienne DSE marquée comme *étant modifiée* est supprimée. Sinon, la marque «*étant modifiée*» est supprimée.

Si le DSA supérieur reçoit une erreur, la modification a échoué. La marque «*étant modifiée*» est supprimée. Si cette modification résulte d'une modification apportée au RDN de l'entrée du préfixe de contexte subordonné, la nouvelle DSE de type **subr**, marquée comme *étant ajoutée* est supprimée. Dans le cas contraire, les mesures qu'il faut prendre n'entrent pas dans le cadre de la présente Spécification d'Annuaire.

En cas de défaillance (par exemple communication du système terminal), le DSA supérieur reprend la procédure à partir de l'étape 2) jusqu'à ce qu'un résultat ou une erreur ait été reçu pour chaque modification en suspens d'un rattachement opérationnel hiérarchique dont il est l'initiateur. Si la modification résulte d'une opération ModifyDN modifiant le RDN de l'entrée du préfixe de contexte subordonné et si le demandeur interrompt l'opération (par exemple en libérant ou en interrompant l'association d'application) avant l'achèvement de la modification, le DSA supérieur ne tient pas compte de cet événement et achève la modification (qui peut aboutir ou non). En pareil cas, l'utilisateur ne sera pas informé du résultat de l'opération ModifyDN.

24.3.2.2 Procédure de modification entreprise par un DSA subordonné

Cette procédure peut être invoquée suite à une intervention administrative (par exemple pour transmettre des modifications aux paramètres **myAccessPoint**, **agreement** ou **valid** du HOB). Par ailleurs, si un DSA subordonné détecte des modifications relatives à la valeur de **SubordinateToSuperior** qu'il a fournie au DSA supérieur, il diffuse la nouvelle information au DSA supérieur en utilisant la procédure ci-après:

- 1) Marquer la DSE de type **cp** comme *étant modifiée*.
- 2) Le DSA subordonné produit une nouvelle valeur de **bindingID** à partir de la valeur existante en augmentant son composant **version**. En utilisant ce nouveau **bindingID**, il envoie au DSA supérieur une opération de modification du rattachement opérationnel avec le paramètre de modification **SubordinateToSuperior**.

9594-4 : 1995 (F)

- 3) Le DSA supérieur vérifie le composant **identifiant** de **bindingID**. S'il n'a pas d'accord de ce genre avec le DSA subordonné, ou si le composant **version** a une valeur inférieure à celle de la version du HOB, il doit retourner une erreur de rattachement opérationnel avec la cause **invalidAgreement**.
- 4) Le DSA supérieur peut accepter la modification du HOB, modifier la DSE représentant la référence subordonnée et retourner un résultat. Sinon, il peut retourner une erreur et mettre fin à l'accord.

De plus, si la DSE supérieure de la DSE (de type **subr**) à renommer est du type **nssr**, le DSA doit, avant de répondre à la demande de modification du HOB) suivre la procédure définie au 19.1.5 (concernant les opérations de modification et les NSSR) afin de garantir que le nouveau nom de l'entrée n'est pas ambigu.

- 5) Si le DSA subordonné reçoit un résultat, la modification est achevée, et il supprime la marque. S'il reçoit une erreur, les mesures qu'il faut prendre n'entrent pas dans le cadre de la présente Spécification d'Annuaire.

En cas de défaillance (par exemple communication du système terminal), Le DSA subordonné reprend la procédure à partir de l'étape 2) jusqu'à ce qu'il ait reçu un résultat ou une erreur pour toutes les opérations de modification en suspens d'un HOB dont il est l'initiateur.

24.3.3 Procédure de terminaison

Les procédures suivantes sont définies pour mettre fin à un HOB qui a été entrepris par la procédure décrite en détail au 24.3.1.

24.3.3.1 Terminaison entreprise par un DSA supérieur

La terminaison d'un rattachement opérationnel hiérarchique n'est entreprise par le DSA supérieur qu'à la suite d'une intervention administrative. Il convient de suivre la procédure ci-après:

- 1) Le DSA supérieur marque la DSE représentant la référence subordonnée comme étant «*supprimée*», de façon que la référence subordonnée ne soit plus utilisée pendant la résolution du nom.
- 2) Le DSA supérieur envoie au DSA subordonné une opération de terminaison de rattachement opérationnel pour le rattachement opérationnel hiérarchique. Le composant **version** de **bindingID** est omis par le DSA supérieur.
- 3) Lorsque le DSA subordonné reçoit la terminaison de rattachement opérationnel, il supprime toute information concernant le rattachement opérationnel hiérarchique et envoie un résultat, à moins que le composant **identifiant** de **bindingID** ne soit pas connu, auquel cas une erreur de rattachement opérationnel avec la cause **invalidID** est retournée. Ce qui se passe pour toute information d'entrée associée au contexte de dénomination subordonné relève d'une initiative locale.
- 4) Si le DSA supérieur reçoit un résultat ou une erreur de rattachement opérationnel avec la cause **invalidID**, il supprime la DSE marquée *supprimée* qui représente la référence subordonnée associée au rattachement opérationnel hiérarchique et supprime toute information concernant le rattachement opérationnel.

En cas de défaillance (par exemple communication du système terminal), le DSA supérieur reprend la procédure à partir de l'étape 2) jusqu'à ce qu'il ait reçu un résultat ou une erreur pour toutes les opérations de terminaison en suspens d'un HOB dont il est l'initiateur.

24.3.3.2 Terminaison entreprise par un DSA subordonné

La terminaison entreprise par le DSA subordonné peut résulter d'une opération de suppression d'entrée qui supprime la dernière entrée dans le contexte de dénomination subordonné, l'entrée du préfixe de contexte, ou bien d'une intervention administrative. Il convient d'appliquer la procédure ci-après:

- 1) le DSA subordonné marque la DSE du préfixe de contexte pour le contexte de dénomination comme étant *supprimée*;
- 2) le DSA subordonné envoie au DSA supérieur une opération terminaison de rattachement opérationnel pour le rattachement opérationnel hiérarchique. Le composant **version** de **bindingID** est omis par le DSA subordonné;
- 3) lorsque le DSA supérieur reçoit la terminaison de rattachement opérationnel, il supprime la DSE qui représente la référence subordonnée associée au rattachement opérationnel et hiérarchique, supprime toute information concernant le rattachement opérationnel et envoie un résultat, à moins que le composant **identifiant** de **bindingID** ne soit pas connu, auquel cas une erreur de rattachement opérationnel est retournée avec la cause **invalidID**;

- 4) si le DSA subordonné reçoit un résultat ou une erreur de rattachement opérationnel avec la cause **invalidID**, il supprime toute information concernant le rattachement opérationnel.

NOTE – Ce qui arrive à l'information d'entrée du contexte de dénomination relève d'une initiative locale au niveau du DSA subordonné. Étant donné que le fait de renommer (par exemple par déplacement) un contexte de dénomination n'est pas autorisé par l'opération **ModifyDN**, un administrateur pourrait, par exemple mettre fin à l'HOB, choisir un autre préfixe de contexte pour le contexte de dénomination et le reconnecter à une autre partie du DIT (c'est-à-dire en établissant un nouvel HOB).

En cas de défaillance (par exemple communication du système terminal), le DSA subordonné reprend la procédure à partir de l'étape 2) jusqu'à ce qu'il ait reçu un résultat ou une erreur pour toutes les opérations de terminaison en suspens d'un HOB dont il est l'initiateur.

24.4 Procédures pour les opérations

Les opérations qui peuvent être exécutées dans l'état coopératif d'un rattachement opérationnel hiérarchique sont définies dans le contexte d'application **directorySystemAC**.

Les procédures que doit suivre le DSA intervenant dans un rattachement opérationnel hiérarchique, sont définies dans les articles 16 à 22.

24.5 Utilisation des contextes d'application

Pour établir, modifier ou terminer un rattachement opérationnel hiérarchique à l'aide du protocole et des procédures de la présente Spécification d'Annuaire, un DSA doit utiliser le contexte d'application **operationalBindingManagementAC**.

25 Rattachement opérationnel hiérarchique non spécifique

Un rattachement opérationnel hiérarchique non spécifique (NHOB) est utilisé pour représenter la relation entre deux DSA détenant deux contextes de dénomination dont l'un est immédiatement subordonné à l'autre. S'agissant d'un NHOB, le DSA supérieur détient une référence subordonnée non spécifique au contexte de dénomination que possède le DSA subordonné; le DSA subordonné détient une référence supérieure immédiate au contexte de dénomination que possède le DSA supérieur. Le rattachement opérationnel garantit que les informations de connaissances appropriées seront échangées et tenues à jour entre les deux DSA de façon que ces deux DSA puissent agir pendant le processus de résolution du nom et d'évaluation des opérations, comme défini dans les articles 18 et 19.

25.1 Caractéristiques typiques du rattachement opérationnel

25.1.1 Symétrie et rôles

Le rattachement opérationnel hiérarchique est du type asymétrique. Dans un rattachement de ce type, les deux rôles sont les suivants:

- a) rôle de DSA maître pour le contexte de dénomination supérieur, *le DSA supérieur* (associé au rôle abstrait «A»); et
- b) le rôle de DSA maître pour le contexte de dénomination subordonné, *le DSA subordonné* (associé au rôle abstrait «B»).

25.1.2 Accord

L'information d'accord échangée pendant l'établissement du rattachement opérationnel hiérarchique non spécifique, qui est une valeur de **NonSpecificHierarchicalAgreement**, contient uniquement le nom distinctif de l'entrée immédiatement supérieure au nouveau contexte de dénomination (composant **immediateSuperior**). Cette information doit être fournie par le DSA qui entreprend le NHOB.

```
NonSpecificHierarchicalAgreement ::= SEQUENCE {  
    immediateSuperior [1] DistinguishedName }
```

NOTE – La façon dont le DSA subordonné détermine que le nom du nouveau contexte de dénomination n'est pas ambigu n'entre pas dans le cadre de la présente Recommandation | Norme internationale. Le nom ne sera pas ambigu s'il est correctement assigné par l'autorité chargée de la dénomination correspondante et si aucun autre DSA ne détient le même nom comme entrée principale.

9594-4 : 1995 (F)

25.1.3 Initiateur**25.1.3.1 Etablissement**

L'établissement d'un rattachement opérationnel hiérarchique non spécifique ne peut être entrepris que par le DSA subordonné. Cette action du DSA subordonné (qui connecte un(e) ou plusieurs entrée(s) ou sous-arbre(s) existant localement au DIT global) est déclenchée par une intervention administrative.

25.1.3.2 Modification

La modification d'un rattachement opérationnel hiérarchique non spécifique peut être entreprise par l'un ou l'autre DSA. Le DSA supérieur peut émettre la modification suite à une modification de l'information du préfixe de contexte supérieur. Cela peut résulter de l'une quelconque des opérations de modification, ou d'une intervention administrative.

Par ailleurs, chaque DSA peut modifier le NHOB si l'information de point d'accès pour son contexte de dénomination (ou l'un de ses contextes de dénomination immédiatement subordonnés dans le cas du DSA subordonné) est modifiée.

25.1.3.3 Terminaison

La terminaison d'un rattachement opérationnel hiérarchique peut être entreprise par l'un ou l'autre DSA. Une telle action de la part du DSA supérieur peut résulter d'une intervention administrative et, dans le cas du DSA subordonné, peut résulter d'une opération de suppression d'entrée qui supprime l'entrée du préfixe de contexte final que détient le subordonné immédiatement subordonné au composant **immediateSuperior** de l'accord, ou bien d'une intervention administrative.

25.1.4 Paramètres d'établissement

Le paramètre d'établissement émis par le DSA supérieur, qui est une valeur de **NHOBSuperiorToSubordinate**, équivaut au paramètre d'établissement de HOB correspondant, mais sans la présence du composant **entryInfo**.

NHOBSuperiorToSubordinate ::= SuperiorToSubordinate (
WITH COMPONENTS { ..., entryInfo ABSENT})

Le paramètre d'établissement émis par le DSA subordonné, qui est une valeur de **NHOBSubordinateToSuperior**, équivaut au paramètre d'établissement de HOB correspondant, mais sans la présence des composants **alias** et **entryInfo**.

NHOBSubordinateToSuperior ::= SubordinateToSuperior (
WITH COMPONENTS { ..., alias ABSENT, entryInfo ABSENT})

25.1.5 Paramètres de modification

Ces paramètres sont identiques aux paramètres d'établissement correspondants et sont utilisés pour mettre en évidence les modifications apportées aux informations fournies dans les paramètres d'établissement après l'établissement du NHOB.

Si un composant quelconque de **NHOBSuperiorToSubordinate** ou de **NHOBSubordinateToSuperior** subit une modification (par exemple le composant **contextPrefixInfo** de **NHOBSuperiorToSubordinate**), le composant correspondant du paramètre de modification (par exemple le composant **contextPrefixInfo** de **NHOBSuperiorToSubordinate**) est fourni dans son intégralité dans la modification du rattachement opérationnel.

25.1.6 Paramètres de terminaison

Aucun des deux DSA ne fournit de paramètre de terminaison en mettant fin à un NHOB.

25.1.7 Identification du type

Le rattachement opérationnel hiérarchique non spécifique est identifié par l'identificateur d'objet assigné lorsque l'on définit l'objet d'information **nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING** (voir 25.2).

25.2 Définition de la classe d'objets d'information de rattachement opérationnel

Ce paragraphe définit le type de rattachement opérationnel hiérarchique non spécifique à l'aide du modèle de la classe d'objets d'information **OPERATIONAL-BINDING** défini dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.

nonSpecificHierarchicalOperationalBinding

OPERATIONAL-BINDING ::=
{

```

AGREEMENT NonSpecificHierarchicalAgreement
APPLICATION CONTEXTS {
    { directorySystemAC }
ASYMMETRIC
    ROLE-A { -- DSA supérieur
        ESTABLISHMENT-PARAMETER      NHOBSuperiorToSubordinate
        MODIFICATION-INITIATOR      TRUE
        MODIFICATION-PARAMETER      NHOBSuperiorToSubordinate
        TERMINATION-INITIATOR      TRUE }
    ROLE-B { -- DSA subordonné
        ESTABLISHMENT-INITIATOR      TRUE
        ESTABLISHMENT-PARAMETER      NHOBSubordinateToSuperior
        MODIFICATION-INITIATOR      TRUE
        MODIFICATION-PARAMETER      NHOBSubordinateToSuperior
        TERMINATION-INITIATOR      TRUE }
ID id-op-binding-non-specific-hierarchical }

```

25.3 Procédures relatives au DSA pour la gestion des rattachements opérationnels non spécifiques

Dans les procédures suivantes, comme pour les procédures décrites au 24.3, une nouvelle marque ou une nouvelle DSE créée par un DSA sera stockée dans une mémoire stable.

Dans la procédure d'établissement et de modification décrite ci-après, le DSA jouant le rôle de celui qui répond (c'est-à-dire n'entretenant pas l'établissement ou la modification) peut fournir au DSA qui joue le rôle initiateur des informations (par exemple attributs opérationnels) qui ne sont pas acceptables pour telle ou telle raison. En pareil cas, le DSA initiateur peut mettre fin au rattachement opérationnel.

25.3.1 Procédure d'établissement

Seul le DSA subordonné peut entreprendre un rattachement opérationnel hiérarchique. Cela pourrait résulter du fait qu'un administrateur souhaite connecter un ou plusieurs sous-arbre(s) d'entrée que possède le DSA à un certain point dans le DIT global. En pareil cas, le DSA subordonné établit un NHOB conformément à la procédure suivante:

- 1) le DSA subordonné a une DSE de type **cp** dans le cadre d'un contexte de dénomination existant ou crée une nouvelle entrée. Il marque la DSE comme *étant ajoutée*, émet un **bindingID** unique et l'enregistre avec la DSE du préfixe de contexte;
- 2) le DSA subordonné envoie une opération d'établissement de rattachement opérationnel au DSA supérieur, contenant les paramètres suivants:
 - a) **bindingType** mis à **nonSpecificHierarchicalOperationalBindingID**;
 - b) paramètre d'établissement **NHOBSubordinateToSuperior**, le cas échéant;
 - c) **NonSpecificHierarchicalAgreement** avec le composant **immediateSuperior** mis au nom distinctif du supérieur immédiat de la nouvelle entrée;
 - d) paramètres **bindingID**, **myAccessPoint** et **valid**, le cas échéant;
- 3) le DSA supérieur vérifie qu'il est le maître pour le supérieur immédiat de la nouvelle entrée du préfixe de contexte ou retourne une erreur de rattachement opérationnel avec la cause **roleAssignment**;
- 4) le DSA supérieur ajoute la DSE du type **nssr** (et l'information d'attribut **nonSpecificKnowledge**) à la DSE du supérieur immédiat de la nouvelle entrée, enregistre **bindingID** avec elle et retourne un résultat;
- 5) si le DSA subordonné reçoit une erreur, il supprime la nouvelle DSE du préfixe de contexte et sa marque. Ce qui arrive à l'information d'entrée à partir de laquelle a été obtenue la DSE du préfixe de contexte relève d'une initiative locale.

Si le DSA subordonné reçoit un résultat, il ajoute les DSE de type **glue**, **subentry**, **admPoint**, **rhob** et **immSupr**, selon le cas, pour représenter l'information **contextPrefixInfo**, et, le cas échéant, une DSE de type **rhob** et **entry** pour représenter l'information **immediateSuperiorInfo**. La marque de la DSE du préfixe de contexte est supprimée.

En cas de défaillance (par exemple communication du système terminal), le DSA subordonné reprend la procédure à partir de l'étape 2) jusqu'à ce qu'il ait reçu un résultat ou une erreur pour toutes les opérations d'établissement en suspens d'un HOB dont il est l'initiateur.

9594-4 : 1995 (F)

25.3.2 Procédure de modification

Si le DSA supérieur détecte des modifications dans l'information **NHOBSuperiorToSubordinate** qu'il a fournie à un DSA subordonné dans un rattachement opérationnel hiérarchique non spécifique, il doit diffuser l'information modifiée au DSA subordonné. Si le NHOB a été établi à l'aide des procédures décrites au 25.3.1, il doit être modifié conformément aux procédures définies pour la modification du HOB au 24.3.2.1 (en substituant **NHOBSuperiorToSubordinate** à **SuperiorToSubordinateModification**).

De même, si le DSA subordonné détecte des modifications dans l'information **NHOBSubordinateToSuperior** qu'il a fournie à un DSA supérieur, il doit diffuser les modifications au DSA supérieur. Si le NHOB a été établi à l'aide des procédures décrites au 25.3.1, il doit être modifié conformément aux procédures définies pour la modification du HOB au 24.3.2.2 (en substituant **NHOBSubordinateToSuperior** à **SubordinateToSuperior**).

25.3.3 Procédure de terminaison

Les procédures suivantes sont définies pour mettre fin à un NHOB qui a été établi à l'aide des procédures décrites au 25.3.1.

25.3.3.1 Terminaison entreprise par le DSA supérieur

La terminaison d'un rattachement opérationnel hiérarchique n'est entreprise par le DSA supérieur qu'à la suite d'une intervention administrative. La procédure ci-après doit être appliquée:

- 1) le DSA supérieur marque comme *étant supprimée* la valeur correspondant au DSA subordonné dans l'attribut **nonSpecificKnowledge** que détient la DSE de l'entrée immédiatement supérieure;
- 2) le DSA supérieur envoie au DSA subordonné une opération de terminaison de rattachement opérationnel pour le NHOB. Le composant **version** de **bindingID** est omis par le supérieur;
- 3) lorsque le DSA subordonné reçoit la terminaison de rattachement opérationnel, il supprime toute information concernant le NHOB et envoie un résultat, à moins que le composant **identifiant** de **bindingID** ne soit inconnu, auquel cas une erreur de rattachement opérationnel est retournée avec la cause **invalidID**. Ce qui arrive à toute information d'entrée associée au contexte de dénomination subordonné relève d'une initiative locale;
- 4) si le DSA supérieur reçoit un résultat ou une erreur de rattachement opérationnel avec la cause **invalidID**, il doit supprimer la valeur de l'attribut **nonSpecificKnowledge** marquée comme *étant supprimé* qui représente l'information de point d'accès associée avec le NHOB, et il supprime toute information concernant le rattachement opérationnel. S'il s'agissait de la dernière valeur de l'attribut **nonSpecificKnowledge**, il supprime l'attribut **nonSpecificKnowledge** et la DSE de type **nssr** dans la DSE.

En cas de défaillance (par exemple communication du système terminal), le DSA supérieur reprend la procédure à partir de l'étape 2) jusqu'à ce qu'il ait reçu un résultat ou une erreur pour toutes les opérations de terminaison en suspens d'un NHOB dont il est l'initiateur.

25.3.3.2 Terminaison entreprise par le DSA subordonné

La terminaison entreprise par le DSA subordonné peut résulter d'une opération de suppression d'entrée qui supprime la dernière entrée dans le contexte de dénomination subordonné (l'entrée du préfixe de contexte) du dernier contexte de dénomination subordonné détenu par le DSA subordonné, ou elle peut résulter d'une intervention administrative. Il convient d'appliquer la procédure ci-après:

- 1) le DSA subordonné marque la DSE du préfixe de contexte de dénomination comme *étant supprimée*;
- 2) le DSA subordonné envoie au DSA supérieur une opération de terminaison de rattachement opérationnel pour le HOB. Le composant **version** de **bindingID** est omis par le subordonné;
- 3) lorsque le DSA supérieur reçoit la terminaison de rattachement opérationnel, il supprime la valeur de l'attribut **nonSpecificKnowledge** qui représente l'information de point d'accès associée avec le NHOB, supprime toute information concernant le rattachement opérationnel, supprime l'attribut **nonSpecificKnowledge** et le type de DSE du type **nssr** dans la DSE immédiatement supérieure au contexte de dénomination subordonné (si la valeur supprimée était la dernière valeur de l'attribut **nonSpecificKnowledge**) et envoie un résultat, à moins que le composant **identifiant** de **bindingID** ne soit inconnu, auquel cas une erreur de rattachement opérationnel est retournée avec la cause **invalidID**;

- 4) si le DSA subordonné reçoit un résultat ou une erreur de rattachement opérationnel avec la cause **invalidID**, il supprime toute information concernant le rattachement opérationnel. Ce qui arrive à toute information d'entrée associée au contexte de dénomination subordonné relève d'une initiative locale.

En cas de défaillance (par exemple communication du système terminal), le DSA subordonné reprend la procédure à partir de l'étape 2) jusqu'à ce qu'il ait reçu un résultat ou une erreur pour toutes les opérations de terminaison en suspens d'un NHOB dont il est l'initiateur.

25.4 Procédures applicables aux opérations

Les opérations qui peuvent être exécutées dans l'état coopératif d'un rattachement opérationnel hiérarchique non spécifique sont définies dans le contexte d'application **directorySystemAC**.

Les procédures que le DSA intervenant dans un rattachement opérationnel hiérarchique non spécifique doit suivre sont définies dans les articles 16 à 22.

25.5 Utilisation des contextes d'application

Pour établir, modifier ou mettre fin à un rattachement opérationnel hiérarchique non spécifique à l'aide du protocole et des procédures de la Spécification d'Annuaire, un DSA doit utiliser le contexte d'application **operationalBindingManagementAC**.

Annexe A

ASN.1 pour les opérations réparties

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

Cette annexe comprend toutes les définitions de type, de valeur et de macro ASN.1 contenues dans la présente Spécification d'Annuaire, sous la forme du module ASN.1 **DistributedOperations**.

DistributedOperations {joint-iso-ccitt ds(5) module(1) distributedOperations(3) 2}

DEFINITIONS ::=

BEGIN

-- EXPORTE TOUT --

-- Les types et les valeurs définis dans ce module sont exportés afin d'être utilisés dans les autres
 -- modules ASN.1 contenus dans les Spécifications d'Annuaire et dans d'autres applications qui
 -- s'en servent pour accéder au service d'Annuaire. D'autres applications pourront les utiliser à
 -- leurs propres fins mais ces utilisations n'obligeront pas à apporter les extensions et les
 -- modifications nécessaires pour tenir à jour ou améliorer le service d'Annuaire.

IMPORTS

**informationFramework, directoryAbstractService, distributedOperations,
 selectedAttributeTypes, directoryOperationalBindingTypes, basicAccessControl, dap**
FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 2}

DistinguishedName, Name, RDNSequence
FROM InformationFramework informationFramework

PresentationAddress, ProtocolInformation, UniqueIdentifier
FROM SelectedAttributeTypes selectedAttributeTypes

AuthenticationLevel
FROM BasicAccessControl basicAccessControl

OPERATION, ERROR
FROM Remote-Operations-Information-Objects
 {joint-iso-ccitt remote-operations(4) informationObjects(5) version1(0) }

**directoryBind, directoryUnbind, read, compare, abandon, list, search, addEntry, removeEntry,
 modifyEntry, modifyDN, referral, OPTIONALLY-SIGNED {}, SecurityParameters**
FROM DirectoryAbstractService directoryAbstractService

id-errcode-dsaReferral
FROM DirectoryAccessProtocol dap

id-op-binding-hierarchical, id-op-binding-non-specific-hierarchical
FROM DirectoryOperationalBindingTypes directoryOperationalBindingTypes ;

-- type paramétrisé pour définir des opérations chaînées --

chained { **OPERATION** : operation } **OPERATION ::=** {
ARGUMENT **OPTIONALLY-SIGNED** { SET {
chainedArgument ChainingArguments,
argument [0] operation.&ArgumentType }}
RESULT **OPTIONALLY-SIGNED** { SET {
chainedResult ChainingResults,
result [0] operation.&ResultType }}
ERRORS { operation.&Errors EXCEPT (referral | dsaReferral) }
CODE operation.&operationCode }

-- opérations de rattachement et de détachement --

dSABind **OPERATION ::=** directoryBind

dSAUnbind **OPERATION ::=** directoryUnbind

-- opérations chaînées --

```

chainedRead OPERATION ::= chained { read }
chainedCompare OPERATION ::= chained { compare }
chainedAbandon OPERATION ::= abandon
chainedList OPERATION ::= chained { list }
chainedSearch OPERATION ::= chained { search }
chainedAddEntry OPERATION ::= chained { addEntry }
chainedRemoveEntry OPERATION ::= chained { removeEntry }
chainedModifyEntry OPERATION ::= chained { modifyEntry }
chainedModifyDN OPERATION ::= chained { modifyDN }

```

-- erreurs et paramètres --

```

dsaReferral ERROR ::= {
    PARAMETER SET {
        reference [0] ContinuationReference,
        contextPrefix [1] DistinguishedName OPTIONAL }
    CODE id-errcode-dsaReferral }

```

-- arguments communs et résultats --

```

ChainingArguments ::= SET {
    originator [0] DistinguishedName OPTIONAL,
    targetObject [1] DistinguishedName OPTIONAL,
    operationProgress [2] OperationProgress
    DEFAULT { nameResolutionPhase notStarted },
    traceInformation [3] TraceInformation,
    aliasDereferenced [4] BOOLEAN DEFAULT FALSE,
    aliasedRDNs [5] INTEGER OPTIONAL,
    -- absent sauf si aliasDereferenced est TRUE
    returnCrossRefs [6] BOOLEAN DEFAULT FALSE,,
    referenceType [7] ReferenceType DEFAULT superior,
    info [8] DomainInfo OPTIONAL,
    timeLimit [9] UTCTime OPTIONAL,
    securityParameters [10] SecurityParameters DEFAULT { },
    entryOnly [11] BOOLEAN DEFAULT FALSE,
    uniqueIdentifier [12] UniqueIdentifier OPTIONAL,
    authenticationLevel [13] AuthenticationLevel OPTIONAL,
    exclusions [14] Exclusions OPTIONAL,
    excludeShadows [15] BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster [16] BOOLEAN DEFAULT FALSE }

```

```

ChainingResults ::= SET {
    info [0] DomainInfo OPTIONAL,
    crossReferences [1] SEQUENCE OF CrossReference OPTIONAL,
    securityParameters [2] SecurityParameters DEFAULT { },
    alreadySearched [3] Exclusions OPTIONAL }

```

```

CrossReference ::= SET{
    contextPrefix [0] DistinguishedName,
    accessPoint [1] AccessPointInformation }

```

```

ReferenceType ::= ENUMERATED {
    superior (1),
    subordinate (2),
    cross (3),
    nonSpecificSubordinate (4),
    supplier(5),
    master (6),
    immediateSuperior (7),
    self (8) }

```

```

TraceInformation ::= SEQUENCE OF TraceItem

```

9594-4 : 1995 (F)

```

TraceItem ::= SET {
    dsa [0] Name,
    targetObject [1] Name OPTIONAL,
    operationProgress [2] OperationProgress }

OperationProgress ::= SET {
    nameResolutionPhase [0] ENUMERATED {
        notStarted (1),
        proceeding (2),
        completed (3) },
    nextRDNTToBeResolved [1] INTEGER OPTIONAL }

DomainInfo ::= ABSTRACT-SYNTAX.&Type

ContinuationReference ::= SET {
    targetObject [0] Name,
    aliasedRDNs [1] INTEGER OPTIONAL,
    operationProgress [2] OperationProgress,
    rdnsResolved [3] INTEGER OPTIONAL,
    referenceType [4] ReferenceType ,
    accessPoints [5] SET OF AccessPointInformation,
    entryOnly [6] BOOLEAN DEFAULT FALSE,
    exclusions [7] Exclusions OPTIONAL,
    returnToDUA [8] BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster [9] BOOLEAN DEFAULT FALSE}

AccessPoint ::= SET {
    ae-title [0] Name,
    address [1] PresentationAddress,
    protocolInformation [2] SET OF ProtocolInformation OPTIONAL }

AccessPointInformation ::= SET {
    COMPONENTS OF MasterOrShadowAccessPoint ,
    additionalPoints [4] SET OF MasterOrShadowAccessPoint OPTIONAL }

MasterOrShadowAccessPoint ::= SET {
    COMPONENTS OF AccessPoint,
    category [3] ENUMERATED {
        master (0),
        shadow (1) } DEFAULT master }

MasterAndShadowAccessPoints ::= SET OF MasterOrShadowAccessPoint

Exclusions ::= SET OF RDNSSequence

```

END

Annexe B

Exemple de résolution répartie du nom

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

L'exemple donné à la Figure B.1 montre comment la résolution répartie du nom est utilisée pour traiter différentes demandes adressées à l'annuaire. L'exemple est fondé sur le DIT hypothétique et sur la ou les configuration(s) correspondante(s) de DSA décrite(s) à l'Annexe M de la Rec. UIT-T X.501 | ISO/CEI 9594-2. Cet arbre est reproduit ci-dessous pour faciliter les consultations.

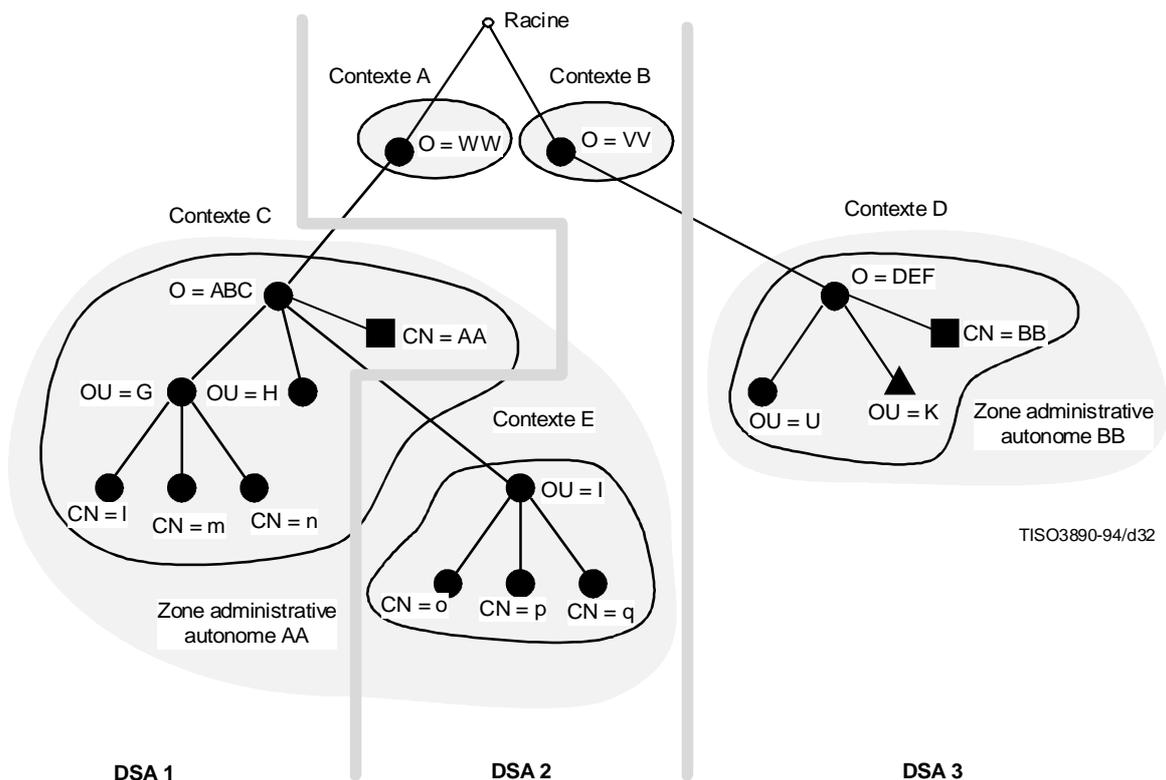


Figure B.1 – DIT hypothétique appliqué à trois DSA

Avec un mode de propagation par chaînage, les demandes suivantes adressées au DSA 1 seront traitées comme suit:

- 1) Une demande avec un nom distinctif {C = WW, O = ABC, OU = G, CN = 1}
 - la résolution du nom assurera la mise en correspondance de chaque RDN dans le nom cible, les DSE étant détenues par le DSA 1, jusqu'à ce que la DSE cible soit localisée.
- 2) Une demande avec un nom distinctif {C = WW, O = JPR}
 - la procédure de résolution du nom dans le DSA 1 mettra en correspondance la DSE C = WW et ne pourra pas assurer d'autres mises en correspondance. A ce stade, le DSA 1 trouve potentiellement deux références qui l'aideront à poursuivre: l'une est la référence **immSupr** dans la DSE C = WW, et l'autre est la référence **supr** dans la DSE racine. Dans cet exemple hypothétique, les deux références donneraient comme indication le DSA 2. En conséquence, la demande est chaînée au DSA 2;

9594-4 : 1995 (F)

- dans le DSA 2, la procédure de résolution du nom assurera la mise en correspondance de la DSE C = WW et ne pourra pas assurer d'autres mises en correspondance. En l'occurrence, étant donné que la DSE C = WW est **cp** et **entry**, et que le DSA 2 est le DSA maître pour cette entrée, et que, par ailleurs, il n'y a pas de nssr à C = WW, le DSA 2 peut déterminer qu'il n'y a pas de tel nom dans l'annuaire. Une erreur de nom noSuchObject est retournée.
- 3) Une demande avec un nom distinctif {C = VV, O = DEF, OU = K}
- la procédure de résolution du nom dans le DSA 1 ne sera pas en mesure d'assurer la mise en correspondance d'une DSE quelconque. La seule référence disponible est la référence **supr** dans la DSE racine, qui donne comme indication le DSA 2. Ainsi, la demande est chaînée au DSA 2;
 - dans le DSA 2, la procédure de résolution du nom assurera la mise en correspondance de la DSE C = VV, puis de la DSE O = DEF, et ne pourra pas assurer d'autres mises en correspondance. Etant donné que la DSE O = DEF est du type **subr**, la référence de connaissances spécifique qui donne comme indication le DSA 3, est utilisée et la demande est chaînée au DSA 3;
 - dans le DSA 3, la procédure de résolution du nom met en correspondance l'intégralité du nom objet cible et constate que la DSE localisée est du type **alias**. En supposant que les alias doivent être déréférencés dans ce cas, on construira un nouveau nom en utilisant le **aliasedEntryName** contenu dans la DSE mise en correspondance. Le DSA 3 reprendra le traitement de la procédure de résolution du nom pour poursuivre.

Annexe C

Utilisation répartie de l'authentification

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

C.1 Résumé

Le modèle de sécurité est défini dans l'article 10 de la Rec. UIT-T X.501 | ISO/CEI 9594-2. Les principales caractéristiques du modèle sont résumées ci-après:

- a) l'authentification simple de l'initiateur de l'opération n'est pas assurée dans le protocole de système d'Annuaire DSP;
- b) l'authentification renforcée, par signature de la demande et du résultat, est assurée dans le DSP;
- c) le chiffrement de la demande ou du résultat n'est pas assuré dans le DSP;
- d) l'authentification des erreurs, y compris des renvois de référence, n'est pas assurée dans le DSP.

La présente annexe décrit la façon dont l'authentification renforcée b) ci-dessus est réalisée dans l'Annuaire réparti. Elle utilise la terminologie et la notation de la Rec. UIT-T X.509 | ISO/CEI 9594-8.

C.2 Authentification simple

L'authentification du DUA fait partie de l'opération de rattachement du protocole d'accès à l'Annuaire DAP. Seul le nom du DUA sera donc véhiculé dans le DSP, dans le champ expéditeur de l'argument de chaînage.

C.3 Modèle d'authentification répartie

La Figure C.1 représente le modèle à utiliser pour spécifier les procédures d'authentification répartie. Ce modèle détermine la séquence des flux d'informations dans le cas général d'une opération de listage ou de recherche. L'opération est considérée comme provenant du DUA «a», qui désigne un objet cible détenu par le DSA «c»; l'exécution de l'opération implique les DSA «b», «c», «d» et «e».

Le DUA «a» se met d'abord en contact avec un DSA quelconque (le DSA «b») qui ne détient pas l'objet cible mais qui peut aller, via un chaînage, vers le DSA (DSA «c») détenant cet objet cible. Si tous les DSA fonctionnent en mode de renvoi de référence, le modèle est notablement simplifié et chaque échange DUA/DSA est équivalent, en termes d'authentification, à l'interaction entre le DUA «a» et le DSA «b».

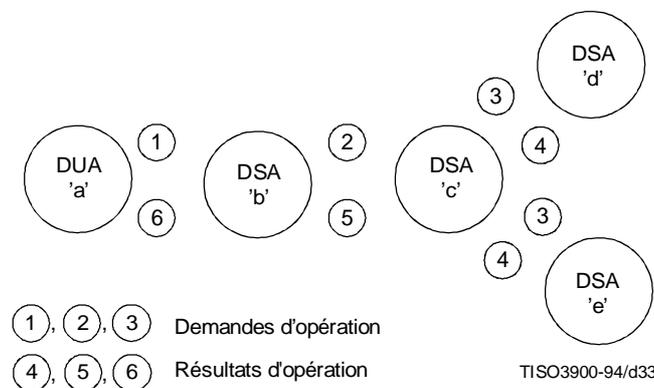


Figure C.1 – Modèle d'authentification répartie

9594-4 : 1995 (F)

C.4 DUA à DSA

L'authentification du demandeur est réalisée suite à l'échange (1) de la Figure C.1; la procédure d'authentification est la suivante:

Soit

OA = l'argument d'opération, c'est-à-dire recherche, lecture, comparaison, etc., tel qu'il est défini dans la Rec. UIT-T X.511 | ISO/CEI 9594-3.

et

a{OA} = l'argument d'opération signé par le DUA «a».

L'authentification sera établie par vérification de la signature.

C.5 Transfert du DAP au DSP

Cette procédure est effectuée par le DSA «b» sur la Figure C.1 et représente le transfert de l'identité signée de l'initiateur du protocole du DAP au DSP.

Le DSA «b» formule l'argument de chaînage approprié, tel qu'il est décrit au 12.3 de la présente Spécification d'Annuaire et le combine avec l'argument d'opération du DAP, formant ainsi une opération chaînée, par exemple: lecture chaînée, recherche chaînée, listage chaîné, etc., du DSP. L'opération chaînée ainsi formée sera signée avant d'être communiquée à d'autres DSA (le DSA «c» sur la Figure C.1). La structure des données pourra être représentée comme suit:

b{ChA, a{OA}} = opération chaînée signée par le DSA «b»

où

ChA = argument de chaînage

Les informations d'authentification véhiculées dans le DSP entre deux DSA (échange (2) sur la Figure C.1) comprennent donc deux parties:

- a) l'argument d'opération signé par l'initiateur, qui permet l'authentification de cet initiateur;
- b) l'opération chaînée, signée par le DSA expéditeur, qui permet l'authentification de ce DSA expéditeur.

C.6 Chaînage au travers de DSA intermédiaires

Cette procédure est effectuée par le DSA «c» dans le modèle décrit à la Figure C.1. Le DSA «c» abandonne la signature fournie par le DSA expéditeur (DSA «b» sur la Figure C.1) et modifie l'argument de chaînage, comme décrit au 10.3. Le DSA «c» doit ensuite combiner l'argument de chaînage modifié avec l'argument d'opération signé, et signer le résultat pour créer une opération chaînée signée, modifiée. Cela peut être représenté par:

c{ChA', a{OA}} = opération chaînée signée par le DSA «c»

où

ChA' = argument de chaînage modifié.

L'opération chaînée modifiée est représentée sur la Figure C.1 par l'échange (3). Selon la nature de l'opération et selon le type de connaissances détenues, le DSA «c» peut effectuer la décomposition de la demande avant le chaînage de toute(s) opération(s) résultante(s). Ceci a été représenté sur la Figure C.1 par le DSA «c», qui envoie des opérations aux DSA «d» et «e»; dans chaque cas, la procédure d'authentification sera identique.

C.7 Authentification des résultats

Le service d'authentification des résultats est demandé par un initiateur d'opération d'Annuaire, à l'aide de l'option **signed** dans le paramètre **protectionRequest Security**. En réponse à une telle demande, un DSA peut (sur option) décider de signer ou non n'importe quel résultat ou tous les résultats: le service d'authentification des résultats n'assure pas l'authentification des réponses erronées.

Lorsqu'un DSA particulier traite les résultats d'un nombre donné de DSA (dont chacun est associé à une demande de service particulière), les cas suivants peuvent se présenter:

- a) Le DSA fournit un ensemble complet de résultats pour une opération, sans qu'il soit nécessaire de les collationner (situation représentée par les DSA «d» et «e» sur la Figure C.1).

- b) Le DSA collationne les résultats locaux (fournis par ce DSA), avec les résultats d'un ou de plusieurs autres DSA (situation représentée par le DSA «c» sur la Figure C.1).
- c) Le DSA chaîne un résultat d'un DSA vers un autre DSA ou DUA et, de ce fait, ne contribue pas à l'ensemble des résultats (situation représentée par le DSA «b» sur la Figure C.1).

C.7.1 Résultats des DSA – Sans collation

Le présent paragraphe traite du rôle d'un DSA quand celui-ci est l'unique source de résultats suite à une demande d'opération particulière, c'est-à-dire lorsqu'il n'a aucune fonction de collation à effectuer. On envisagera ci-dessus le cas du DSP et celui du DAP.

C.7.1.1 DSP

Le DSA peut choisir d'exécuter l'une des procédures suivantes:

- a) retourner les résultats non signés, ce qui peut être représenté par:
 $ChR, OR =$ résultat d'opération chaînée (non signé)
où:
 $ChR =$ résultats de chaînage
 $OR =$ résultat d'opération
- b) signer uniquement le résultat d'opération, ce qui peut être représenté par:
 $ChR, d\{OR\} =$ résultats d'opération signés par le DSA «d»
- c) signer uniquement le résultat d'opération chaînée, ce qui peut être représenté par:
 $d\{ChR, OR\} =$ résultat d'opération chaînée signé par le DSA «d»
- d) signer le résultat d'opération et le résultat d'opération chaînée, ce qui peut être représenté par:
 $d\{ChR, d\{OR\}\} =$ résultat d'opération et résultat d'opération chaînée, signés par le DSA «d».

NOTE – Dans le cas où le résultat d'opération est signé, il sera retourné par l'initiateur; dans le cas où le résultat d'opération chaînée a été signé, le DSA destinataire devra abandonner la signature pour pouvoir modifier l'argument chaining results avant de transmettre le résultat d'opération chaînée.

C.7.1.2 DAP

La procédure étant intégralement décrite dans la Rec. UIT-T X.511 | ISO/CEI 9594-3, ce qui suit est un résumé, fourni afin de disposer d'un texte complet.

Le DSA peut choisir de retourner les résultats non signés, ce qui peut être représenté par:

- $OR =$ résultat d'opération
- ou signés, ce qui peut être représenté par:
 $d\{OR\} =$ résultat d'opération signé par le DSA «d».

C.7.2 Résultats de protocole de DSA – Avec collation

Le présent paragraphe décrit le rôle d'un DSA qui renvoie le résultat d'une demande de service particulière, lorsque la collation et l'intégration des résultats provenant d'autres DSA sont une condition préalable nécessaire. On considérera ci-dessous le cas du DSP et celui du DAP.

C.7.2.1 DSP

Considérant que zéro ou plusieurs résultats provenant d'autres DSA peuvent être signés cette procédure permet à un DSA de collationner et d'intégrer les résultats, de ne signer aucune partie constitutive du résultat composite ou d'en signer plusieurs et, facultativement de signer l'ensemble du résultat composite.

C.7.2.1.1 Production de l'argument résultats de chaînage

Cette procédure exige qu'un DSA (représenté par le DSA «c» sur la Figure C.1) supprime des résultats provenant des DSA extérieurs (les DSA «d» et «e» sur la Figure C.1) toutes les signatures de résultat d'opérations chaînées. Le DSA «c» possède alors un ensemble de résultats de chaînage non signés, un ensemble de résultat d'opération signés et un ensemble de résultat d'opération non signés.

9594-4 : 1995 (F)

Tous les résultats de chaînage sont traités comme décrit dans la présente Spécification d'Annuaire, pour créer un résultat de chaînage unique modifié, désigné par:

- i) ChR' = résultats de chaînage modifiés.

C.7.2.1.2 Résultats obtenus localement, non signés

Si le DSA ne désire pas signer les résultats obtenus localement, l'ensemble des résultat d'opération non signés est fusionné avec le résultat local pour former un ensemble modifié de résultat d'opération, désigné par:

OR' = résultat d'opération fusionné.

L'ensemble complet des résultat d'opération sera donc l'union de l'ensemble des résultat d'opération signés à l'extérieur, désigné par:

$d\{OR\}, e\{OR\} \dots$

et de l'ensemble résultat d'opération fusionné, union qui sera désignée par:

- ii) $OR', d\{OR\}, e\{OR\} \dots$ = résultat d'opération.

C.7.2.1.3 Résultats obtenus localement, signés

Si le DSA désire signer les résultats obtenus localement, les résultat d'opération non signés provenant de l'extérieur sont d'abord fusionnés. L'ensemble complet des résultats d'opération est donc l'union de l'ensemble des résultats d'opération signés localement, désigné par OR'' , et de l'ensemble des résultats d'opération signés à l'extérieur, désigné par:

$d\{OR\}, e\{OR\}, \dots$, union qui sera désignée par:

- iii) $c\{OR\}, OR'', d\{OR\}, e\{OR\}, \dots$ = résultat d'opération.

C.7.2.1.4 Résultat d'opération chaînée non signée

Si le DSA ne désire pas signer le résultat d'opération chaînée, ce dernier comprendra les résultats de chaînage [spécifiés au point i) ci-dessus] ajoutés au résultat d'opération [spécifié aux points ii) ou iii) ci-dessus], union qui sera désignée par:

$ChR', OR', d\{OR\}, e\{OR\}, \dots$ = résultat d'opération chaînée (non signé)

ou par:

$ChR', c\{OR\}, OR'', d\{OR\}, e\{OR\}, \dots$ = résultat d'opération chaînée (non signé) et résultat d'opération signé par le DSA «c».

C.7.2.1.5 Résultat d'opération chaînée signé

Si le DSA désire signer le résultat d'opération chaînée, ce résultat comprendra les résultats de chaînage [spécifiés au point i) ci-dessus] ajoutés au résultat d'opération [spécifié aux points ii) ou iii) ci-dessus], union qui sera désignée par:

$c\{ChR', OR', d\{OR\}, e\{OR\}, \dots\}$ = résultat d'opération chaînée signé par le DSA «c»

ou par:

$c\{ChR', c\{OR\}, OR, d\{OR\}, e\{OR\}, \dots\}$ = résultat d'opération chaînée et résultat d'opération signé par le DSA «c».

C.7.2.2 DAP

La procédure est très similaire à celle qui est décrite au C.7.2.1, sauf que l'argument résultats de chaînage n'est pas transféré dans le DAP.

C.7.3 Résultats chaînés du DSA

Ce paragraphe traite des procédures que doit effectuer un DSA pour chaîner un résultat d'opération en retour vers le DSA ou le DUA demandeur, respectivement dans le DSP et dans le DAP.

C.7.3.1 DSP

Le DSA commence par supprimer la signature (s'il en existe une) du résultat d'opération chaînée. Puis il traite l'argument résultats de chaînage comme décrit dans la présente Spécification d'Annuaire pour produire un argument résultats de chaînage modifiés. Ce dernier est alors fusionné avec l'argument résultat d'opération pour produire un résultat d'opération chaînée modifié. En dernier lieu, le DSA peut, facultativement, signer le résultat d'opération chaînée avant de le communiquer au DSA suivant de la chaîne.

C.7.3.2 DAP

Un DSA (représenté par le DSA «b» de la Figure C.1) supprime d'abord la signature (s'il en existe une) du résultat d'opération chaînée. Puis il analyse et abandonne l'argument résultats de chaînage et enfin signe, facultativement, l'argument résultat d'opération restant, avant de communiquer le résultat au DUA.

Annexe D

Spécification des types de rattachement opérationnel hiérarchique et de rattachement opérationnel hiérarchique non spécifique

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

Elle comprend les définitions des classes d'objets d'information contenues dans la présente Spécification d'Annuaire sous la forme du module ASN.1 **HierarchicalOperationalBindings**.

HierarchicalOperationalBindings

{joint-iso-ccitt ds(5) module(1) hierarchicalOperationalBindings(20) 2}

DEFINITIONS ::=

BEGIN

-- EXPORTE TOUT --

-- Les types et les valeurs définis dans ce module sont exportés afin d'être utilisés dans les autres
-- modules ASN.1 contenus dans les Spécifications d'Annuaire et dans d'autres applications qui
-- s'en serviront pour accéder au service d'Annuaire. D'autres applications pourront les utiliser à
-- leurs propres fins mais ces utilisations n'obligeront pas à apporter les extensions et les
-- modifications nécessaires pour tenir à jour ou améliorer le service d'Annuaire.

IMPORTS

informationFramework, distributedOperations, directoryOperationalBindingTypes,
opBindingManagement, dsp

FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 2}

Attribute, RelativeDistinguishedName, DistinguishedName

FROM InformationFramework informationFramework

MasterAndShadowAccessPoints

FROM DistributedOperations distributedOperations

directorySystemAC

FROM DirectorySystemProtocol dsp

OPERATIONAL-BINDING

FROM OperationalBindingManagement opBindingManagement

id-op-binding-hierarchical, id-op-binding-non-specific-hierarchical

FROM DirectoryOperationalBindingTypes directoryOperationalBindingTypes ;

-- types --

HierarchicalAgreement ::= SEQUENCE {
 rdn [0] RelativeDistinguishedName,
 immediateSuperior [1] DistinguishedName }

NonSpecificHierarchicalAgreement ::= SEQUENCE {
 immediateSuperior [1] DistinguishedName }

SuperiorToSubordinate ::= SEQUENCE {
 contextPrefixInfo[0] DITcontext,
 entryInfo [1] SET OF Attribute OPTIONAL,
 immediateSuperiorInfo [2] SET OF Attribute OPTIONAL }

DITcontext ::= SEQUENCE OF Vertex

Vertex ::= SEQUENCE {
 rdn [0] RelativeDistinguishedName,
 admPointInfo [1] SET OF Attribute OPTIONAL,
 subentries [2] SET OF SubentryInfo OPTIONAL,
 accessPoints [3] MasterAndShadowAccessPoints OPTIONAL }

SubentryInfo ::= SEQUENCE {
 rdn [0] RelativeDistinguishedName,
 info [1] SET OF Attribute }

```

SubordinateToSuperior ::= SEQUENCE {
    accessPoints [0] MasterAndShadowAccessPoints OPTIONAL,
    alias [1] BOOLEAN DEFAULT FALSE,
    entryInfo [2] SET OF Attribute OPTIONAL }

SuperiorToSubordinateModification ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT})

NHOBSuperiorToSubordinate ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT })

NHOBSubordinateToSuperior ::= SubordinateToSuperior (
    WITH COMPONENTS { ..., alias ABSENT, entryInfo ABSENT })

-- objets informationnels de rattachement opérationnel --

hierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT HierarchicalAgreement
    APPLICATION CONTEXTS {
        { directorySystemAC }}
    ASYMMETRIC
    ROLE-A { -- DSA supérieur --
        ESTABLISHMENT-INITIATOR TRUE
        ESTABLISHMENT-PARAMETER SuperiorToSubordinate
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER SuperiorToSubordinateModification
        TERMINATION-INITIATOR TRUE }
    ROLE-B { -- DSA inférieur --
        ESTABLISHMENT-INITIATOR TRUE
        ESTABLISHMENT-PARAMETER SubordinateToSuperior
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER SubordinateToSuperior
        TERMINATION-INITIATOR TRUE }
    ID id-op-binding-hierarchical }

nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT NonSpecificHierarchicalAgreement
    APPLICATION CONTEXTS {
        { directorySystemAC }}
    ASYMMETRIC
    ROLE-A { -- DSA supérieur --
        ESTABLISHMENT-PARAMETER NHOBSuperiorToSubordinate
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER NHOBSuperiorToSubordinate
        TERMINATION-INITIATOR TRUE }
    ROLE-B { -- DSA inférieur --
        ESTABLISHMENT-INITIATOR TRUE
        ESTABLISHMENT-PARAMETER NHOBSubordinateToSuperior
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER NHOBSubordinateToSuperior
        TERMINATION-INITIATOR TRUE }
    ID id-op-binding-non-specific-hierarchical }

END

```

Annexe E

Exemple d'administration des connaissances

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

La présente annexe illustre l'administration des connaissances (définie à l'article 23) avec un exemple unique. La Figure ci-dessous contient les symboles utilisés pour décrire les arbres d'information de DSA de cinq DSA.

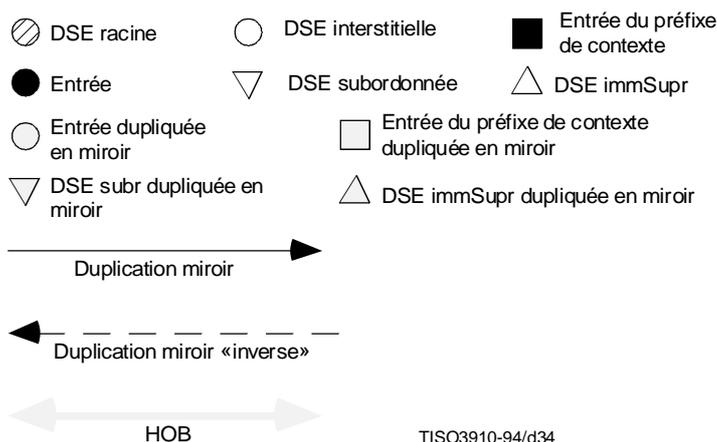


Figure E.1 – Symboles utilisés pour décrire les arbres d'information de DSA

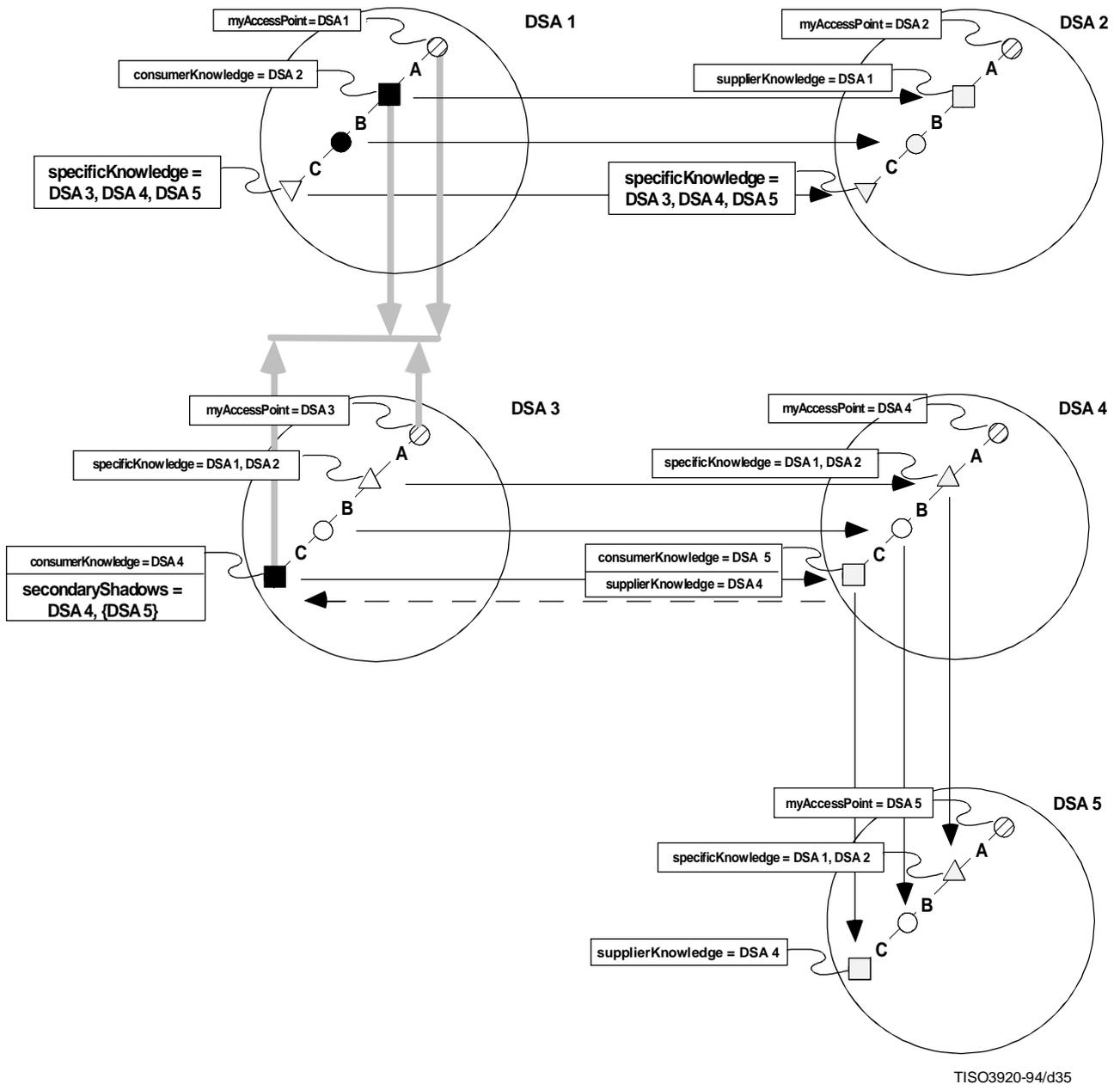
Dans la Figure E.2, le DSA 1 est le maître pour le contexte de dénomination {A}, avec les deux entrées {A} et {A, B}. Le DSA 1 détient une référence subordonnée pour le contexte de dénomination {A, B, C}, dont l'administration est assurée via un HOB avec le DSA 3. Le DSA 1 est un fournisseur d'informations miroirs pour le DSA 2: il lui transmet les copies de l'information d'utilisateur du contexte de dénomination {A} et la référence subordonnée au contexte de dénomination {A, B, C} qui identifie les points d'accès des DSA 3, DSA 4 et DSA 5, le premier étant le maître pour le contexte de dénomination subordonné.

Le DSA 3 est le maître pour le contexte de dénomination {A, B, C}. Cet agent détient non seulement l'entrée unique {A, B, C} du contexte de dénomination, mais encore une référence supérieure immédiate pour le contexte de dénomination {A} dont l'administration est assurée via un HOB avec le DSA 1. Le DSA 3 est un fournisseur d'informations miroirs pour le DSA 4: il lui transmet les copies de l'information d'utilisateur du contexte de dénomination {A, B, C} et la référence supérieure immédiate au contexte de dénomination {A} qui identifie les points d'accès des DSA 1 et DSA 2, ce dernier étant le maître pour le contexte de dénomination supérieure. Le DSA 4 est un fournisseur d'informations miroirs (secondaires) pour le DSA 5, auquel il transmet une copie de l'information qu'il reçoit de la part du DSA 3.

La Figure E.2 illustre les attributs opérationnels de DSA utilisés pour représenter les connaissances et en assurer l'administration.

Le DSA 1 utilise la valeur de son attribut **myAccessPoint** (associé avec sa DSE racine) et les valeurs d'usage commun de son attribut **consumerKnowledge** (associé au préfixe de contexte {A}) pour former une valeur du type **MasterAndShadowAccessPoints** destinée à être utilisée dans les interactions de HOB avec le DSA 3. A son tour, le DSA 3 utilise la valeur de son attribut **myAccessPoint** (associé avec sa DSE racine) et les valeurs d'usage commun de son attribut **consumerKnowledge** ainsi que de son attribut **secondaryShadows** (tous deux associés au préfixe de contexte {A, B, C}) pour former une valeur du type **MasterAndShadowAccessPoints** destinée à être utilisée dans ses interactions de HOB avec le DSA 1. Ensemble, les deux DSA utilisent le DOP pour assurer l'administration d'une référence subordonnée détenue par le DSA 1 et une référence supérieure immédiate détenue par le DSA 3. La référence

subordonnée du DSA 1, exprimée par un attribut **specificKnowledge** associé avec une DSE à {A, B, C} est fondée sur la valeur **MasterAndShadowAccessPoints** qu'elle reçoit de la part du DSA 3; la référence supérieure immédiate du DSA 3, exprimée par un attribut **specificKnowledge** associé avec une DSE à {A}, est également fondée sur la valeur **MasterAndShadowAccessPoints** qu'elle reçoit de la part du DSA 1.



TISO3920-94/d35

Figure E.2 – Exemple d'administration des connaissances

9594-4 : 1995 (F)

Les DSA 1 et DSA 2 utilisent leurs valeurs de **myAccessPoint** dans les interactions de rattachement opérationnel de duplication miroir pour tenir à jour une valeur de **consumerKnowledge** dans le DSA 1 (identifiant le point d'accès du DSA 2 et de **supplierKnowledge** dans le DSA 2 (identifiant le point d'accès du DSA 1, les deux attributs étant associés au préfixe de contexte {A}). Ensemble les deux DSA utilisent le DOP pour tenir à jour la référence du consommateur détenue par le DSA 1 et la référence du fournisseur détenue par le DSA 2.

Le DSA 2 reçoit une copie de l'attribut **specificKnowledge** associé au préfixe de contexte {A, B, C} de la part du DSA 1 dans les interactions de DISP avec le DSA 1. Ces interactions servent à tenir à jour la référence subordonnée du DSA 2 au préfixe de contexte {A, B, C}.

Les DSA 3 et DSA 4 (et, par la même occasion, les DSA 4 et DSA 5) tiennent à jour les références du consommateur et du fournisseur, respectivement, de manière analogue à l'interaction qui existe entre les DSA 1 et DSA 2.

Le DSA 4 reçoit une copie de l'attribut **specificKnowledge** associé au préfixe de contexte {A4} de la part du DSA 3 dans les interactions de DISP avec le DSA 3. Ces interactions servent à tenir à jour la référence supérieure immédiate du DSA 4 au préfixe de contexte {A}.

Le DSA 4 communique au DSA 3 toutes modifications apportées à ses attributs **myAccessPoint** et **consumerKnowledge** (et de l'attribut **secondaryShadows**, qui est nul dans cet exemple) en utilisant l'opération de modification de rattachement opérationnel du DOP. Le DSA 4 fournit au DSA 3 une valeur de l'ensemble **SuppliersAndConsumers**, contenant uniquement les valeurs de l'attribut **consumerKnowledge** qui identifient les points d'accès des DSA qui détiennent des duplications miroirs d'usage commun; les valeurs de l'attribut **secondaryShadows** fournies par le DSA 4, selon le cas, seraient toutes intrinsèquement utilisables de manière générale (dans cet exemple, le DSA 5 est supposé détenir une copie d'usage commun du contexte de dénomination à {A, B, C}). Le DSA 3 utilise cette information pour tenir à jour une valeur de son attribut **secondaryShadows** associé au préfixe de contexte {A, B, C}. Cet attribut, comme indiqué ci-dessus, est utilisé dans les interactions de DOP avec le DSA 1 pour tenir à jour la référence subordonnée du DSA 1 au préfixe de contexte {A, B, C}.

Le DSA 5 tient à jour sa référence supérieure immédiate au préfixe de contexte {A} en utilisant les interactions de DISP avec le DSA 4, de manière analogue aux interactions qui existent entre les DSA 3 et DSA 4.

Annexe F

Amendements et correctifs

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

La présente version de la Spécification d'Annuaire comprend les amendements suivants:

- Amendement 1 pour le contrôle d'accès;
- Amendement 2 pour la duplication, le schéma et la recherche améliorée.

La présente version de la Spécification d'Annuaire comprend les correctifs (et projets de correctif) suivants qui corrigent les défauts signalés dans les relevés de défauts ci-après:

- Corrigendum technique 1 (reprenant les relevés de défauts 004, 010-013, 022, 023, 025-027, 029).
- Corrigendum technique 2 (reprenant les relevés de défauts 002, 034, 048, 050, 059).
- Corrigendum technique 3 (reprenant les relevés de défauts 024, 062, 065, 066).