



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**X.952**

(12/97)

SÉRIE X: RÉSEAUX POUR DONNÉES ET  
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Traitement réparti ouvert

---

**Technologies de l'information – Traitement  
réparti ouvert – Fonction de courtage:  
fourniture de la fonction de courtage au  
moyen du service d'annuaire OSI**

Recommandation UIT-T X.952

(Antérieurement Recommandation du CCITT)

---

RECOMMANDATIONS UIT-T DE LA SÉRIE X

**RÉSEAUX POUR DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS**

<b>RÉSEAUX PUBLICS POUR DONNÉES</b>	
Services et fonctionnalités	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
<b>INTERCONNEXION DES SYSTÈMES OUVERTS</b>	
Modèle et notation	X.200–X.209
Définitions 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
Tests de conformité	X.290–X.299
<b>INTERFONCTIONNEMENT DES RÉSEAUX</b>	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.399
<b>SYSTÈMES DE MESSAGERIE</b>	X.400–X.499
<b>ANNUAIRE</b>	X.500–X.599
<b>RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES</b>	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
<b>GESTION OSI</b>	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
<b>SÉCURITÉ</b>	X.800–X.849
<b>APPLICATIONS OSI</b>	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
<b>TRAITEMENT RÉPARTI OUVERT</b>	<b>X.900–X.999</b>

*Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.*

## **NORME INTERNATIONALE 13235-3**

### **RECOMMANDATION UIT-T X.952**

#### **TECHNOLOGIES DE L'INFORMATION – TRAITEMENT RÉPARTI OUVERT – FONCTION DE COURTAGE: FOURNITURE DE LA FONCTION DE COURTAGE AU MOYEN DU SERVICE D'ANNUAIRE OSI**

#### **Résumé**

La présente Recommandation | Norme internationale contient la description de la façon dont la fonction de courtage ODP peut être réalisée au moyen d'entrées informationnelles et de mécanismes supports de l'annuaire OSI. Elle doit être utilisée conjointement avec la norme relative à la fonction de courtage ODP (voir la Rec. UIT-T X.950 | ISO/CEI 13235-1). En cas de désaccord entre les prescriptions de la Recommandation UIT-T X.950 et celles de la présente Spécification, ce sont les prescriptions de la Recommandation UIT-T X.950 qui l'emportent.

La présente Spécification porte sur:

- les gabarits normalisés pour les objets informationnels de la fonction de courtage dans l'arbre DIT;
- la description du mappage des opérations de la fonction de courtage sur des opérations appropriées de l'annuaire;
- la description de l'utilisation d'autres caractéristiques de l'annuaire qui permettent de fournir les mécanismes supports pour la réalisation de la fonction de courtage ODP.

La présente Spécification vise à permettre la réalisation de la fonction de courtage ODP au moyen de l'annuaire OSI, lorsque c'est nécessaire.

#### **Source**

La Recommandation X.952 de l'UIT-T a été approuvée le 12 décembre 1997. Un texte identique est publié comme Norme internationale ISO/CEI 13235-3.

## 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. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT.

Dans certains secteurs des technologies 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.

## 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.

## DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 1998

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

## TABLE DES MATIÈRES

		<i>Page</i>
1	Domaine d'application.....	1
2	Références normatives .....	1
	2.1 Recommandations   Normes internationales identiques.....	1
3	Définitions.....	2
4	Abréviations.....	4
5	Aperçu général.....	4
6	Schéma.....	5
6.1	Généralités.....	6
6.2	Entrée courtier .....	7
	6.2.1 commonName.....	7
	6.2.2 traderInterface.....	8
	6.2.3 dsaName.....	8
	6.2.4 typeRepos.....	8
	6.2.5 defSearchCard.....	8
	6.2.6 maxSearchCard.....	8
	6.2.7 defMatchCard.....	9
	6.2.8 maxMatchCard.....	9
	6.2.9 defReturnCard.....	9
	6.2.10 maxReturnCard.....	9
	6.2.11 defHopCount.....	10
	6.2.12 maxHopCount.....	10
	6.2.13 defFollowPolicy.....	10
	6.2.14 maxFollowPolicy.....	11
	6.2.15 maxLinkFollowPolicy.....	11
	6.2.16 supportsModifiableProperties.....	11
	6.2.17 supportsDynamicProperties.....	11
	6.2.18 supportsProxyOffers.....	12
	6.2.19 maxList.....	12
	6.2.20 requestIdStem.....	12
	6.2.21 description.....	12
	6.2.22 userPassword.....	12
	6.2.23 Autres attributs X.500.....	12
6.3	Entrée politiques du courtier.....	13
	6.3.1 commonName.....	13
	6.3.2 typeManagementConstraint.....	13
	6.3.3 searchConstraint.....	14
	6.3.4 offerAcceptanceConstraint.....	14
	6.3.5 Autres attributs X.500.....	14
6.4	Entrée offre de service.....	14
	6.4.1 sOfferId.....	15
	6.4.2 serviceInterfaceId.....	16
	6.4.3 serviceTypeId.....	16
	6.4.4 hasDynamicProperties.....	16
	6.4.5 hasModifiableProperties.....	17
	6.4.6 dynamicProps.....	17
	6.4.7 Autres attributs X.500.....	17
6.5	Entrée lien du courtier.....	18
	6.5.1 linkName.....	18
	6.5.2 linkId.....	18
	6.5.3 targetTraderInterfaceId.....	19
	6.5.4 defPassOnFollowRule.....	19
	6.5.5 limitingFollowRule.....	19
	6.5.6 Autres attributs X.500.....	19

	<i>Page</i>	
6.6	Entrée offre de délégation .....	20
6.6.1	proxyOfferId .....	20
6.6.2	proxyLookUpInterfaceId .....	21
6.6.3	constraintRecipe.....	21
6.6.4	ifMatchAll.....	21
6.6.5	Autres attributs X.500.....	21
6.7	Autres entrées X.500 utilisées par l'agent T-DUA .....	22
7	Opérations .....	22
7.1	Initialisation .....	23
7.2	Opérations du client .....	23
7.3	Opérations de registre .....	23
7.3.1	Exportation .....	24
7.3.2	Retrait .....	25
7.3.3	Modification .....	25
7.3.4	Description.....	26
7.3.5	Retrait avec contrainte .....	27
7.3.6	Résolution .....	27
7.4	Opérations de consultation.....	28
7.4.1	Opération d'interrogation .....	28
7.4.2	Politiques .....	28
7.4.3	Recherche locale .....	29
7.4.4	Recherche parmi des courtiers fédérés.....	30
7.4.5	Recherche d'offres de délégation .....	30
7.4.6	Offres de service renvoyées .....	30
7.5	Opérations relatives aux liens .....	31
7.5.1	Adjonction de lien.....	31
7.5.2	Suppression de lien .....	31
7.5.3	Modification de lien.....	32
7.5.4	Description de lien .....	32
7.5.5	Listage des liens.....	33
7.6	Opérations relatives aux offres de délégation .....	33
7.6.1	Exportation d'offre de délégation.....	33
7.6.2	Retrait d'offre de délégation.....	34
7.6.3	Description d'offre de délégation .....	34
7.7	Opérations relatives aux attributs du courtier .....	35
7.8	Opérations administratives.....	35
7.8.1	Listage des offres .....	35
7.8.2	Listage des offres de délégation.....	36
7.9	Opérations d'évaluation de propriété dynamique .....	36
7.9.1	Evaluation de propriété dynamique .....	37
8	Répertoire de types.....	37
8.1	Schéma X.500 et répertoire de types minimal .....	37
9	Propriétés dynamiques .....	38
9.1	Exportation d'une offre de service .....	38
9.2	Importation d'une offre de service .....	39
	Annexe A – Définition schématique des définitions de courtier.....	40
	Annexe B – Exemple de définition schématique de description de service.....	50

## Introduction

La fonction de courtage ODP (voir la Rec. UIT-T de la série X.950 | ISO/CEI 13235) permet d'offrir un service et de découvrir les services qui ont été offerts. La Rec. UIT-T X.950 | ISO/CEI 13235-1 contient la définition d'une spécification d'entreprise, d'une spécification d'information et d'une spécification de traitement de cette fonction de courtage. Aucune spécification d'ingénierie n'est définie dans la Rec. UIT-T X.950 | ISO/CEI 13235-1. La présente Recommandation | Norme internationale donne la description de la façon dont la fonction de courtage, Rec. UIT-T X.950 | ISO/CEI 13235-1, peut être réalisée au moyen du service d'annuaire OSI (voir la Rec. UIT-T X.500 | ISO/CEI 9594-1) concernant le stockage des informations et la fourniture de mécanismes supports. La présente Spécification n'impose pas qu'un courtier soit réalisé au moyen de l'annuaire OSI. Mais si l'annuaire OSI est utilisé, la présente Spécification contient la définition de gabarits normalisés pour les entrées informationnelles (par exemple objets informationnels offre de service et lien) de l'arbre DIT de l'annuaire.

L'article 5 donne un aperçu général de la manière dont la fonction de courtage est réalisée sous forme de combinaison d'un agent DUA et d'un agent DSA X.500. On utilise l'agent DSA X.500 pour stocker l'objet informationnel courtier et un agent d'utilisateur d'annuaire de type courtier (T-DUA) réalise la fonctionnalité requise par un courtier, qui est difficile, voire impossible, à réaliser au moyen des services d'annuaire OSI.

L'article 6 définit les gabarits normalisés pour les entrées informationnelles de l'objet informationnel courtier, à savoir les informations qu'un courtier donné connaît.

L'article 7 décrit le mappage des opérations de la fonction de courtage sur des opérations appropriées de l'annuaire.

L'article 8 spécifie une fonction de répertoire de type minimal qui est nécessaire afin de permettre le fonctionnement correct de l'annuaire X.500 pour le courtage.

L'article 9 décrit les mécanismes utilisés pour permettre le traitement des propriétés dynamiques des offres de service d'un courtier.

La présente Spécification contient deux annexes.

L'Annexe A (normative) contient une définition de schéma pour les définitions relatives aux courtiers.

L'Annexe B (normative) contient une définition de schéma relative à un exemple de description de service.



## NORME INTERNATIONALE

## RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – TRAITEMENT RÉPARTI OUVERT –  
FONCTION DE COURTAGE: FOURNITURE DE LA FONCTION  
DE COURTAGE AU MOYEN DU SERVICE D'ANNUAIRE OSI**

## 1 Domaine d'application

La présente Spécification contient la description de la façon dont la fonction de courtage ODP peut être réalisée au moyen d'entrées informationnelles et de mécanismes supports de l'annuaire OSI. Elle doit être utilisée conjointement avec la norme relative à la fonction de courtage ODP (Rec. UIT-T X.950 | ISO/CEI 13235-1). En cas de désaccord entre les prescriptions de la Rec. UIT-T X.950 | ISO/CEI 13235-1 et celles de la présente Spécification, ce sont les prescriptions de la Rec. UIT-T X.950 | ISO/CEI 13235-1 qui l'emportent.

La présente Spécification porte sur:

- les gabarits normalisés pour les objets informationnels de la fonction de courtage figurant dans l'arbre DIT;
- la description du mappage des opérations de la fonction de courtage sur des opérations appropriées de l'annuaire;
- la description de l'utilisation d'autres caractéristiques de l'annuaire qui permettent de fournir les mécanismes supports pour la réalisation de la fonction de courtage ODP.

La présente Spécification n'impose pas qu'un courtier soit réalisé au moyen de l'annuaire OSI. Mais si l'annuaire OSI est utilisé, la présente Spécification contient la définition de gabarits normalisés pour les entrées informationnelles (par exemple objets informationnels offre de service et lien) de l'arbre DIT de l'annuaire. Elle n'impose aucune restriction quant aux endroits où ces entrées sont placées dans l'arbre DIT de l'annuaire. Autrement dit, elle ne donne la normalisation d'aucune règle de structure. Elle donne simplement la description d'un mécanisme permettant de réaliser la fonction de courtage au moyen de l'annuaire OSI.

La présente Spécification vise à permettre la réalisation de la fonction de courtage ODP au moyen de l'annuaire OSI, lorsque c'est nécessaire.

## 2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations 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 tient à jour une liste des Recommandations de l'UIT-T en vigueur.

### 2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.500 (1993) | ISO/CEI 9594-1:1995, *Technologies 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 9594-2:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: les modèles* .
- Recommandation UIT-T X.509 (1993) | ISO/CEI 9594-8:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: cadre d'authentification* .

- Recommandation UIT-T X.511 (1993) | ISO/CEI 9594-3:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: définition du service abstrait*.
- Recommandation UIT-T X.519 (1993) | ISO/CEI 9594-5:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: spécification du protocole*.
- Recommandation UIT-T X.520 (1993) | ISO/CEI 9594-6:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés*.
- Recommandation UIT-T X.521 (1993) | ISO/CEI 9594-7:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: classes d'objets sélectionnées*.
- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels*.
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes*.
- Recommandation UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un*.
- Recommandation UIT-T X.902 (1995) | ISO/CEI 10746-2:1996, *Technologies de l'information – Traitement ouvert réparti – Modèle de référence: fondements*.
- Recommandation UIT-T X.903 (1995) | ISO/CEI 10746-3:1996, *Technologies de l'information – Traitement ouvert réparti – Modèle de référence: architecture*.
- Recommandation UIT-T X.950 (1997) | ISO/CEI 13235-1:1998, *Technologies de l'information – Traitement réparti ouvert – Fonction de courtage: spécification*.

### 3 Définitions

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.902 | ISO/CEI 10746-2:

- activité;
- comportement;
- objet client;
- défaillance;
- identificateur;
- instance;
- interaction;
- interface;
- signature d'interface;
- nom;
- objet;
- obligation;
- système ODP;
- politique;
- objet serveur;
- sous-type;
- gabarit de <X>;
- courtage;

- type;
- point de vue.

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.903 | ISO/CEI 10746-3:

- administrateur;
- communauté;
- point de vue traitement;
- référence d'interface d'ingénierie;
- point de vue ingénierie;
- point de vue entreprise;
- exportateur;
- importateur;
- point de vue information;
- exportation de service;
- importation de service;
- offre de service;
- point de vue technologie;
- fonction de courtage;
- fonction de répertoire de types.

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.950 | ISO/CEI 13235-1:

- courtiers fédérés;
- itérateur;
- lien;
- offre de délégation;
- type de service;
- propriété de service;
- courtier;
- attribut de courtier;
- graphe de courtage.

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.500 | ISO/CEI 9594-1:

- annuaire;
- base de données d'annuaire;
- utilisateur (d'annuaire).

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- attribut;
- type d'attribut;
- valeur d'attribut;
- arbre d'informations d'annuaire;
- agent de système d'annuaire;
- agent d'utilisateur d'annuaire;
- nom distinctif;
- entrée (d'annuaire);
- filtre;

- règle de concordance;
- nom (d'annuaire);
- forme de nom;
- objet;
- classe d'objets;
- entrée d'objet;
- nom distinctif relatif;
- règle de structure;
- sous-classe;
- subordonné;
- superclasse.

La présente Recommandation | Norme internationale utilise les opérations suivantes, qui sont définies dans la Rec. UIT-T X.511 | ISO/CEI 9594-3:

- adjonction d'entrée;
- modification d'entrée;
- lecture;
- suppression d'entrée;
- recherche.

La présente Recommandation | Norme internationale utilise les termes suivants, qui sont définis dans la Rec. UIT-T X.509 | ISO/CEI 9594-8:

- authentification;
- mot de passe.

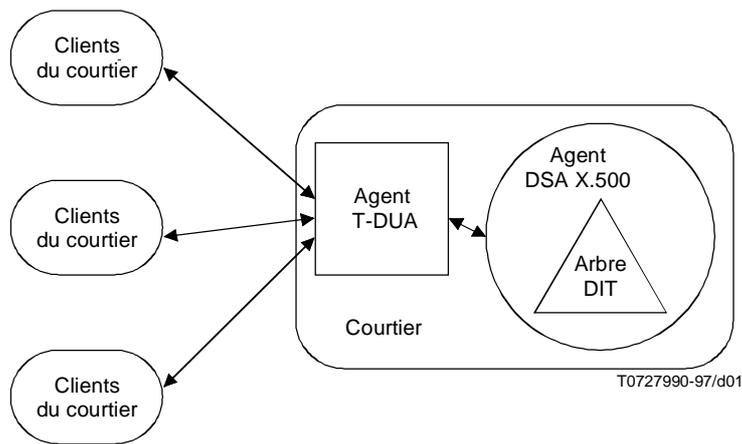
## 4 Abréviations

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

DIB	Base de données d'annuaire ( <i>directory information base</i> )
DIT	Arbre d'informations de l'annuaire ( <i>directory information tree</i> )
DN	Nom distinctif ( <i>distinguished name</i> )
DSA	Agent de système d'annuaire ( <i>directory system agent</i> )
DUA	Agent d'utilisateur d'annuaire ( <i>directory user agent</i> )
ODP	Traitement réparti ouvert ( <i>open distributed processing</i> )
OID	Identificateur d'objet ( <i>object identifier</i> )
RDN	Nom distinctif relatif ( <i>relative distinguished name</i> )
T-DUA	Agent d'utilisateur d'annuaire de type courtier ( <i>trader directory user agent</i> )

## 5 Aperçu général

Dans la présente Spécification, la fonction de courtage est réalisée sous forme de combinaison d'un agent DUA et d'un agent DSA X.500. Dans la mesure du possible, on utilise les caractéristiques de l'annuaire X.500 pour réaliser directement la fonction de courtage, mais les caractéristiques de courtier ne peuvent pas toutes être directement réalisées au moyen de l'annuaire X.500. C'est pourquoi le courtier (l'objet qui assure la fonction de courtage) possède deux composantes: un agent DSA X.500 utilisé pour stocker les informations de courtier et un agent d'utilisateur d'annuaire de type courtier (T-DUA) réalisant la fonctionnalité requise par un courtier, qui est difficile, voire impossible, à réaliser au moyen de l'annuaire X.500. On utilise l'agent DSA X.500 pour stocker l'objet informationnel courtier. Les demandes émanant des clients du courtier (importateurs et exportateurs) sont mappées sur des opérations relatives à la base de données X.500. La Figure 1 montre les composantes d'un courtier et ses interactions avec ses clients.



**Figure 1 – Le courtier, ses composantes et ses clients**

L'agent T-DUA et les clients du courtier (importateurs et exportateurs) communiquent au moyen d'un protocole de courtage. Ce protocole n'est pas défini dans la présente Spécification. Il peut s'agir de n'importe quel protocole qui met en œuvre la fonctionnalité spécifiée dans la Rec. UIT-T X.950 | ISO/CEI 13235-1. La présente Spécification vise à spécifier comment l'agent T-DUA utilise un agent DSA X.500 pour prendre en charge la fonctionnalité spécifiée dans la Rec. UIT-T X.950 | ISO/CEI 13235-1.

Les informations stockées par l'agent DSA X.500 comprennent:

- les attributs du courtier (c'est-à-dire les informations sur le courtier);
- les politiques d'entreprise du courtier (c'est-à-dire les règles permettant de déterminer et de guider le comportement du courtier);
- l'ensemble des offres de service (c'est-à-dire les informations utilisées par le courtier lorsqu'il joue le rôle de serveur);
- l'ensemble des liens du courtier (c'est-à-dire les informations utilisées par le courtier lorsqu'il joue le rôle de client);
- l'ensemble des offres de délégation (c'est-à-dire les informations utilisées par le courtier lorsqu'il joue le rôle de serveur pour les offres de délégation).

On utilise un agent DSA X.500 pour stocker ces informations pour plusieurs raisons:

- le modèle informationnel requis par le courtier ODP est très semblable à celui de l'annuaire X.500;
- l'annuaire X.500 offre une souplesse importante, en autorisant la définition de nouveaux attributs X.500 au moment de l'exécution;
- il est justifié de mettre à profit l'investissement existant relatif à l'annuaire X.500 plutôt que d'essayer de créer une infrastructure entièrement nouvelle;
- le courtier peut ainsi utiliser l'infrastructure X.500 générale pour consulter les adresses de présentation de courtiers et clients liés et employer les caractéristiques de sécurité de l'annuaire X.500 pour authentifier les utilisateurs.

NOTE – Les détails concernant la façon de fournir l'infrastructure X.500 et les caractéristiques de sécurité de l'annuaire X.500 pour la fonction de courtage sortent du cadre de la présente Spécification.

Il est impossible de réaliser un courtier ODP entièrement au moyen de l'annuaire X.500 car le modèle opérationnel utilisé par le courtier ODP présente des différences importantes, parmi lesquelles:

- les opérations du courtier pour lesquelles il n'existe pas de projection directe sur des opérations X.500;
- les opérations réparties qui sont mises en œuvre au moyen des informations stockées dans les liens et attributs de courtier dont la signification n'est pas la même que pour la répartition mise en œuvre dans le cadre de l'annuaire X.500.

## 6 Schéma

Ce schéma X.500 donne la description de la portion d'un arbre DIT X.500 servant à stocker les informations qu'un courtier particulier connaît. Ce schéma, qui est fondé sur le modèle de l'annuaire X.500, est donné dans l'Annexe A.

### 6.1 Généralités

Les informations qu'un courtier donné connaît (l'objet informationnel courtier) sont gardées dans un sous-arbre de l'arbre DIT X.500. Ce sous-arbre peut être rattaché n'importe où dans l'arbre DIT global et aucune règle de structure n'est définie pour contrôler sa position. Le sous-arbre de courtier devrait normalement être rattaché sous les entrées d'organisation et d'unité organisationnelle (représentant respectivement les informations connues par les courtiers de l'organisation et d'une unité organisationnelle). Les informations connues par des courtiers distincts sont conservées séparément dans l'arbre DIT et on ne cherche nullement à mapper le modèle de répartition utilisé dans le cadre de l'annuaire X.500 au modèle de répartition très différent des courtiers fédérés.

Les informations de courtier sont stockées dans l'arbre DIT X.500 sous la forme de parties indépendantes. Chaque partie contient les informations connues par un seul courtier. Dans l'exemple de la Figure 2, deux courtiers sont représentés: un pour l'organisation dans son ensemble et un autre pour une unité de cette organisation. Les établissements de liens entre ces deux courtiers se font via le protocole de courtage et non via le protocole X.500.

L'objet informationnel courtier (voir la Figure 3) est composé de cinq types d'entrées:

- l'entrée courtier, qui contient des renseignements sur le courtier;
- l'entrée politiques du courtier, qui contient des renseignements sur les politiques d'entreprise du courtier;
- les entrées offre de service, qui contiennent des renseignements sur les offres de service connues par le courtier;
- les entrées lien du courtier, qui contiennent des renseignements sur les liens avec les autres courtiers;
- les entrées offre de délégation, qui contiennent des renseignements sur les offres de délégation connues par le courtier.

NOTE 1 – La structure des offres de service, liens et offres de délégation représentée sur la Figure 3 n'est qu'un exemple possible de structure d'information.

NOTE 2 – Outre les attributs X.500 énumérés pour chaque entrée, la présence d'autres attributs dans une entrée ne constitue pas une transgression de la présente Spécification. D'autres attributs X.500 peuvent être nécessaires pour les raisons suivantes:

- si une application donnée de courtage nécessite d'autres attributs X.500 spécifiques, ceux-ci peuvent être définis dans la spécification de cette application de courtage;
- si une réalisation donnée de courtier nécessite d'autres attributs X.500 spécifiques, ceux-ci peuvent être définis dans la documentation relative à cette réalisation.

D'autres attributs peuvent être inclus sous la forme de classes d'objets auxiliaires.

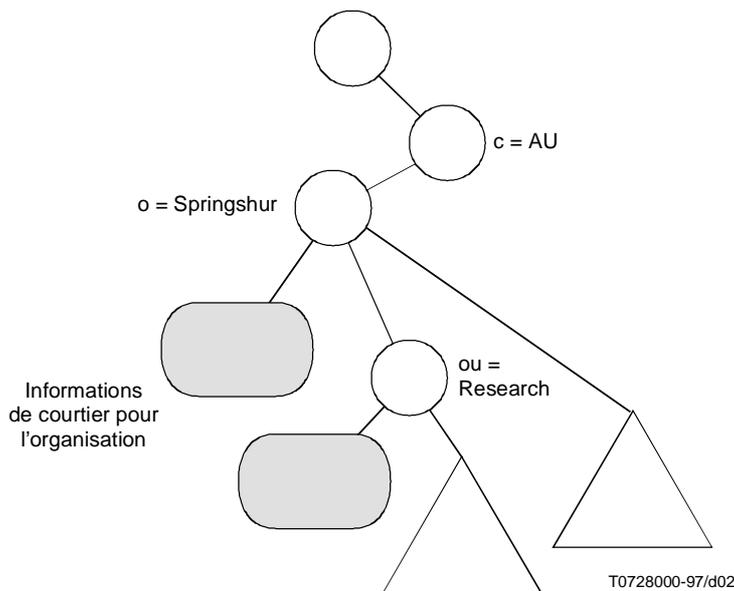


Figure 2 – Exemple de deux courtiers stockés dans l'arbre DIT X.500

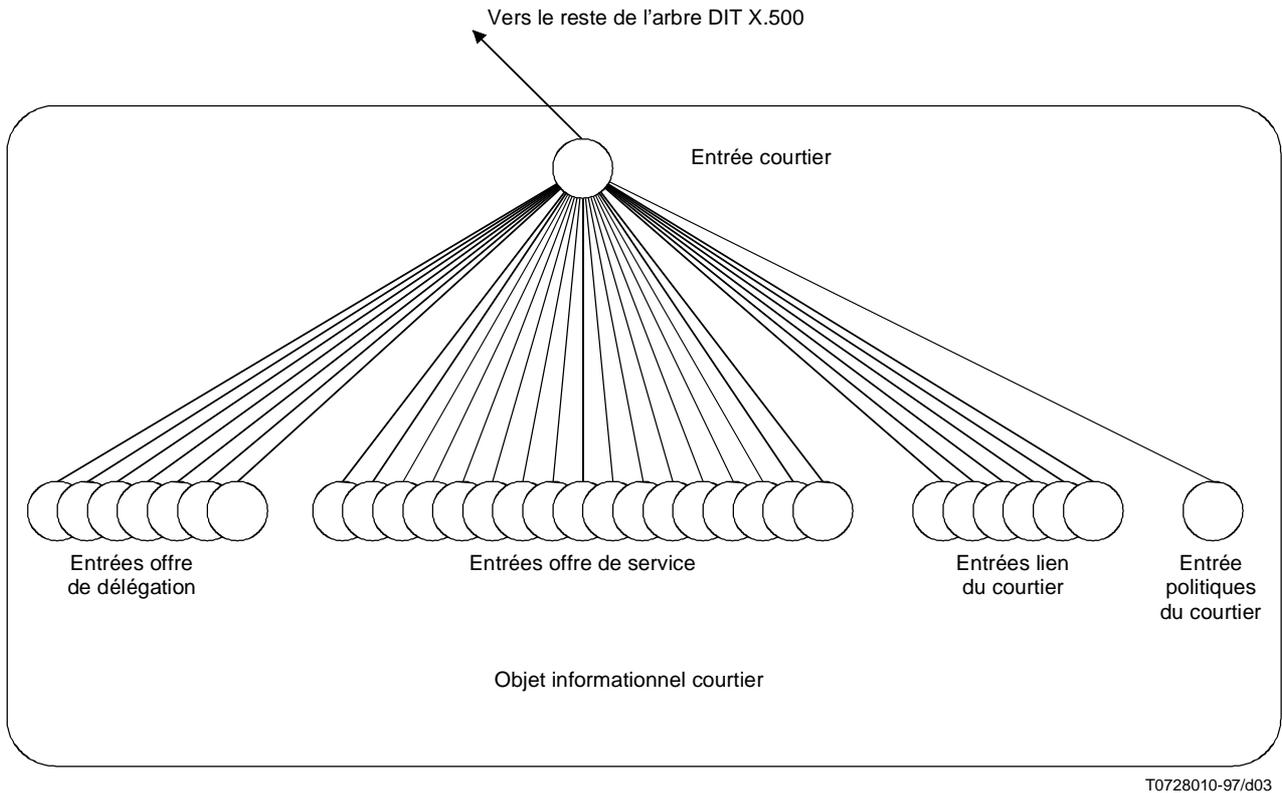


Figure 3 – Exemple d'objet informationnel courtier avec cinq types d'entrées X.500

## 6.2 Entrée courtier

La racine du sous-arbre courtier est constituée par l'entrée courtier. Cette entrée, qui contient des informations sur le courtier (attributs du courtier – politiques de courtage et caractéristiques du courtier normalisées), est utilisée comme une donnée de configuration par le courtier (agent T-DUA) au moment de son initialisation. Ces informations sont exprimées sous la forme d'un ensemble d'attributs X.500 qui représentent les attributs du courtier.

```
traderEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {commonName | traderInterface | dsaName | typeRepos |
                        defSearchCard | maxSearchCard | defMatchCard |
                        maxMatchCard | defReturnCard | maxReturnCard |
                        defHopCount | maxHopCount | defFollowPolicy |
                        maxFollowPolicy | maxLinkFollowPolicy |
                        supportsModifiableProperties | supportsDynamicProperties
                        | supportsProxyOffers | maxList | requestIdStem}
    MAY CONTAIN          {description | userPassword}
    ID                   id-trader-oc-traderEntry}
```

### 6.2.1 commonName

Il s'agit du nom du courtier. L'attribut `commonName` constitue le nom RDN de l'entrée courtier. Le nom complet d'un courtier correspond au nom distinctif de cette entrée (c'est-à-dire le nom d'accès complet de l'entrée courtier dans l'ensemble de l'arbre DIT X.500). Le nom distinctif complet identifie de manière univoque ce courtier parmi l'ensemble des courtiers de l'annuaire X.500. Il s'agit d'un attribut X.500 normalisé, défini dans la Rec. UIT-T X.520 | ISO/CEI 9594-6.

### 6.2.2 traderInterface

Il s'agit de l'adresse du courtier, c'est-à-dire de l'adresse de présentation à laquelle ce courtier peut être contacté. Cet attribut X.500 est utilisé par le courtier comme partie de sa donnée de configuration au moment de l'initialisation ainsi que par d'autres courtiers lorsqu'ils souhaitent répartir l'importation d'un courtier parmi des courtiers fédérés.

```
traderInterface ATTRIBUTE ::= {
    SUBTYPE OF          presentationAddress
    SINGLE VALUE        TRUE
    ID                  id-trader-at-traderInterface}
```

### 6.2.3 dsaName

Il s'agit du nom de l'agent DSA associé à l'objet courtier.

```
dsaName ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-dsaName}
```

### 6.2.4 typeRepos

Il s'agit du nom de répertoire de types utilisé par le courtier pour la conservation des définitions de types de service, types d'interface et types de propriétés de service.

```
typeRepos ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-typeRepos}
```

### 6.2.5 defSearchCard

Il s'agit du nombre maximal par défaut des offres de service à examiner avant qu'il ne soit mis fin à une recherche. Cette valeur est utilisée si aucune valeur n'est spécifiée par un importateur. Elle ne doit pas dépasser la valeur de l'attribut maxSearchCard.

```
defSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defSearchCard }
```

### 6.2.6 maxSearchCard

Il s'agit du nombre maximal des offres de service qu'un courtier examine avant de mettre fin à une recherche donnée.

```
maxSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxSearchCard}
```

**6.2.7 defMatchCard**

Il s'agit du nombre maximal par défaut des offres concordantes trouvées avant qu'un courtier ne mette fin à une recherche. Cette valeur est utilisée si aucune valeur n'est spécifiée par un importateur. Elle ne doit pas dépasser la valeur de l'attribut maxMatchCard.

```
defMatchCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-defMatchCard}
```

**6.2.8 maxMatchCard**

Il s'agit du nombre maximal des offres concordantes trouvées avant qu'un courtier ne mette fin à une recherche donnée.

```
maxMatchCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-maxMatchCard}
```

**6.2.9 defReturnCard**

Il s'agit du nombre maximal par défaut des offres de service renvoyées à un importateur. Cette valeur est utilisée si aucune valeur n'est spécifiée par un importateur. Elle ne doit pas dépasser la valeur de l'attribut maxReturnCard.

```
defReturnCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-defReturnCard}
```

**6.2.10 maxReturnCard**

Il s'agit du nombre maximal des offres de service renvoyées à un importateur.

```
maxReturnCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-maxReturnCard}
```

**6.2.11 defHopCount**

Il s'agit de la profondeur maximale par défaut des liens à traverser avant qu'il ne soit mis fin à une recherche. Cette valeur est utilisée si aucune valeur n'est spécifiée par un importateur. Elle ne doit pas dépasser la valeur de l'attribut maxHopCount.

```
defHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-defHopCount }
```

**6.2.12 maxHopCount**

Il s'agit de la profondeur maximale des liens à traverser avant qu'il ne soit mis fin à une recherche donnée.

```
maxHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-maxHopCount }
```

**6.2.13 defFollowPolicy**

Il s'agit du comportement de poursuite de lien par défaut lorsque aucun comportement de poursuite de lien n'est spécifié par un importateur. Le comportement de poursuite associé à un lien peut avoir l'une des valeurs suivantes:

- local\_only – jamais de poursuite sauf déclaration explicite dans une opération;
- if\_no\_local – poursuite uniquement en l'absence d'offre locale concordante;
- always – poursuite dans tous les cas sauf si certaines politiques l'interdisent.

```
defFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-defFollowPolicy }
```

```
FollowOption ::= ENUMERATED{
    localOnly    (0),
    ifNoLocal   (1),
    always      (2)}
```

**6.2.14 maxFollowPolicy**

Il s'agit du comportement de poursuite limite applicable à tous les liens d'un courtier pour une interrogation donnée. Il peut avoir la priorité sur les politiques de lien tout comme celles d'importateur correspondant au comportement de poursuite de lien.

```
maxFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxFollowPolicy }
```

**6.2.15 maxLinkFollowPolicy**

Il s'agit du comportement de poursuite le moins contraignant pour un lien au moment de la création ou de la modification du comportement limite pour un lien quelconque d'un courtier.

```
maxLinkFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxLinkFollowPolicy }
```

**6.2.16 supportsModifiableProperties**

S'il a la valeur 'true' (vrai), cet attribut permet d'invoquer l'opération de modification d'offre.

```
supportsModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsModifiableProperties }
```

**6.2.17 supportsDynamicProperties**

S'il a la valeur 'true' (vrai), cet attribut permet aux offres de service d'avoir des propriétés dynamiques, c'est-à-dire des propriétés dont il faut obtenir les valeurs lorsque c'est nécessaire pour une mise en concordance ou une description d'offres de service ou pour une mise en concordance d'offres de délégation.

```
supportsDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsDynamicProperties }
```

**6.2.18 supportsProxyOffers**

S'il a la valeur 'true' (vrai), cet attribut permet l'exportation, le retrait, la description et le listage d'offres de délégation, qui sont des offres de service spéciales permettant de déterminer en cours d'exécution l'interface à laquelle le service signalé est fourni.

```
supportsProxyOffers ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE    booleanMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-supportsProxyOffers }
```

**6.2.19 maxList**

Il s'agit de la dimension maximale de la liste d'un itérateur que le courtier souhaite prendre en charge.

```
maxList ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-maxList }
```

**6.2.20 requestIdStem**

Il s'agit d'une identification du courtier, à utiliser comme radical pour la production d'un identificateur dans le cadre d'une demande d'interrogation qui est lancée par ce courtier et qui doit être transmise à un courtier lié.

```
requestIdStem ATTRIBUTE ::= {
    WITH SYNTAX                OCTET STRING (SIZE (0..ub-request-id-stem))
    EQUALITY MATCHING RULE    octetStringMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-requestIdStem }
```

**6.2.21 description**

Il s'agit d'une description textuelle du courtier. La description est un champ de texte libre décrivant le courtier (par exemple "CSIRO courtier de la division des technologies de l'information"). Il s'agit d'un attribut X.500 normalisé défini dans la Rec. UIT-T X.520 | ISO/CEI 9594-6.

**6.2.22 userPassword**

Il s'agit d'un mot de passe permettant une authentification simple lorsque l'agent T-DUA accède à l'objet informationnel courtier. Les règles de contrôle d'accès relatives à cet attribut doivent être telles que cet attribut ne soit pas lisible dans le cas général. Il s'agit d'un attribut X.500 normalisé défini dans la Rec. UIT-T X.509 | ISO/CEI 9594-8.

**6.2.23 Autres attributs X.500**

D'autres attributs X.500 requis par une réalisation ou une application donnée peuvent être inclus sous la forme de classes d'objets auxiliaires. Donnons quelques exemples d'autres attributs X.500 pouvant figurer dans cette entrée:

- information de contrôle d'accès (sur l'entrée courtier);
- information de contact pour l'administrateur du courtier;
- information de contact humain pour le service, y compris une description textuelle du service;
- information de limite (par exemple la quantité maximale de ressources pouvant être consommées par une interrogation, la durée de vie maximale d'une offre).

### 6.3 Entrée politiques du courtier

L'entrée politiques du courtier est située immédiatement sous l'entrée courtier dans le sous-arbre d'information. Elle contient les politiques d'entreprise du courtier (politiques définies dans la spécification d'entreprise de la fonction de courtage figurant dans la Rec. UIT-T X.950 | ISO/CEI 13235-1), exprimées sous la forme d'un ensemble de contraintes. Chaque contrainte peut à son tour s'exprimer sous la forme d'une chaîne décrivant la règle, qui consiste en un filtre X.500 composé d'attributs de l'entrée courtier, ou sous la forme du nom d'un objet qui met en œuvre la politique en question.

```
traderPolicyEntryNF NAME-FORM ::= {
    NAMES                traderPolicyEntry
    WITH ATTRIBUTES      {commonName}
    ID                   id-trader-nf-traderPolicy}

traderPolicyEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {commonName }
    MAY CONTAIN          {typeManagementConstraint | searchConstraint |
                        offerAcceptanceConstraint }
    ID                   id-trader-oc-traderPolicy}
```

Les contraintes associées aux politiques sont définies comme suit:

```
PolicySpecification ::= CHOICE {
    stringRule           [0]   DirectoryString{ub-policy-string-rule}
    policyObjectId       [1]   DistinguishedName }

policySpecificationMatch MATCHING-RULE ::= {
    SYNTAX               PolicySpecification
    ID                   id-trader-mr-policySpecificationMatch}

-- La règle renvoie TRUE (vrai) si deux spécifications contiennent exactement les mêmes
-- caractères.
```

#### 6.3.1 commonName

L'attribut commonName constitue le nom RDN de l'entrée politiques du courtier et possède la valeur 'Trader Policies' (politiques du courtier). Il s'agit d'un attribut X.500 normalisé défini dans la Rec. UIT-T X.520 | ISO/CEI 9594-6.

#### 6.3.2 typeManagementConstraint

Cette contrainte est une règle relative à la spécification des types et aux relations entre les types.

```
typeManagementConstraint ATTRIBUTE ::= {
    WITH SYNTAX          PolicySpecification
    EQUALITY MATCHING RULE policySpecificationMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-typeManagementConstraint}
```

### 6.3.3 searchConstraint

Cette contrainte est une règle guidant la recherche d'offres appropriées dans le système du courtier, par exemple, recherche séquentielle ou recherche en parallèle parmi des courtiers fédérés.

```
searchConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-searchConstraint}
```

### 6.3.4 offerAcceptanceConstraint

Cette contrainte est une règle qui restreint l'ensemble des offres de service acceptables pour le courtier. Par exemple, un courtier peut n'accepter que les offres de service présentant certains types de service.

```
offerAcceptanceConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-offerAcceptanceConstraint}
```

### 6.3.5 Autres attributs X.500

D'autres attributs X.500 requis par une réalisation ou une application donnée peuvent être inclus sous la forme de classes d'objets auxiliaires. Donnons quelques exemples d'autres attributs X.500 pouvant figurer dans cette entrée:

- information de contrôle d'accès (sur l'entrée politiques du courtier);
- autres contraintes associées aux politiques d'entreprises du courtier.

## 6.4 Entrée offre de service

Les entrées offre de service (zéro, une ou plusieurs) sont situées immédiatement sous l'entrée courtier dans le sous-arbre d'information. Chaque entrée offre de service contient les renseignements relatifs à une offre de service et une seule. L'ensemble des entrées offre de service constituent l'ensemble des offres de service. Chaque entrée offre de service comporte une classe d'objets structurelle serviceOfferEntry et exactement une classe d'objets auxiliaire parmi celles qui sont énumérées dans la règle de contenu applicable à cette entrée. Chaque classe d'objets auxiliaire correspond à un type de service et donne la définition des propriétés de service qui sont obligatoires et de celles qui sont facultatives. Chaque entrée offre de service contient donc les attributs obligatoires et facultatifs relatifs à cette entrée ainsi que les attributs obligatoires et facultatifs déterminés par la classe d'objets auxiliaire correspondante, qui spécifie les propriétés obligatoires et facultatives du service. Chaque entrée offre de service est dénommée au moyen de l'attribut sOfferId de l'entrée.

```
serviceOfferEntryNF NAME-FORM ::= {
    NAMES                      serviceOfferEntry
    WITH ATTRIBUTES            {sOfferId}
    ID                          id-trader-nf-serviceOffer}
serviceOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF                {top}
    MUST CONTAIN                {sOfferId | serviceInterfaceId | serviceTypeId
                                hasDynamicProperties | hasModifiableProperties }
    MAY CONTAIN                 {dynamicProps}
    ID                          id-trader-oc-serviceOffer}
```

Les propriétés de service d'une offre de service sont stockées sous la forme d'attributs X.500. Les attributs particuliers (propriétés de service obligatoires) qui doivent être présents dans une offre de service et ceux qui y sont autorisés (propriétés de service facultatives) sont contrôlés par le schéma X.500. Les noms de propriété se projettent sur des types d'attribut X.500 et les valeurs de propriété sur des valeurs d'attribut X.500. Les noms de propriété sont des identificateurs d'objet. Les propriétés de service sont contrôlées par la classe d'objets auxiliaire associée au type de service de l'offre de service. En outre, des règles de contrôle d'accès peuvent rendre non modifiable par les utilisateurs (c'est-à-dire en lecture seule) toute propriété d'une offre de service.

A chaque identificateur de type de service sont associés un type d'interface et un type de propriétés de service qui peuvent être obtenus à partir du répertoire de types. L'exportateur doit spécifier un identificateur de type de service valide connu par le courtier. Représenté sous la forme d'une classe d'objets auxiliaire X.500, l'identificateur de type de propriétés de service associé est donc stocké dans chaque entrée offre de service sous la forme d'une valeur de l'attribut de classe d'objets. Cette classe d'objets auxiliaire donne la définition des attributs X.500 obligatoires et facultatifs (propriétés de service) qui doivent/pouvent figurer dans une offre de service de ce type de service.

Les classes d'objets auxiliaires peuvent être déduites d'autres classes d'objets X.500 (on parle de formation de sous-classes dans la X.500). Cela revient à définir un type de propriétés de service à partir de la définition d'un autre type de propriétés de service. Pour bien comprendre les règles relatives à la formation de sous-classes, il faut se reporter à la liste des attributs obligatoires et à la liste des attributs facultatifs:

- la sous-classe doit contenir tous les attributs de la liste des attributs obligatoires de sa superclasse;
- la sous-classe peut contenir n'importe quel attribut de la liste des attributs facultatifs de sa superclasse;
- une classe d'objets peut être une sous-classe de plusieurs autres classes. La sous-classe doit contenir tous les attributs de toutes les listes des attributs obligatoires de toutes ses superclasses et peut contenir n'importe quel attribut parmi les attributs des listes des attributs facultatifs de ses superclasses.

NOTE – Dans la définition des entrées offre de service, la classe d'objets structurelle d'une telle entrée correspond à l'offre de service générique, tandis que le type de propriétés de service qui donne une spécialisation de l'offre de service en lui attribuant un type de service particulier constitue une classe d'objets auxiliaire. Il aurait été possible de faire de l'offre de service une classe d'objets abstraite, les divers types de propriétés de service auraient alors constitué une classe d'objets structurelle déduite de la classe abstraite d'offre de service. Ceci présente l'inconvénient de nécessiter un schéma X.500 complexe avec de nombreuses règles.

Pour une offre donnée, un exportateur peut aussi choisir d'attribuer une valeur dynamique à une propriété donnée. En d'autres termes, la valeur de la propriété n'est pas spécifiée au moment de l'exportation mais elle doit être évaluée lorsque c'est nécessaire pour une mise en concordance ou une description d'offres de service (par exemple, par l'utilisation des opérations d'interrogation ou de description relatives aux courtiers). En revanche, l'exportateur spécifie l'interface à laquelle la propriété peut être évaluée.

#### 6.4.1 sOfferId

Il s'agit de l'identificateur de l'entrée offre de service qui est attribué par le courtier. Il constitue le nom RDN de l'entrée offre de service. Cet identificateur doit être unique parmi tous les identificateurs d'offre de service détenus par ce courtier. L'identificateur sOfferId fait partie de l'identificateur d'offre de service que l'agent T-DUA renvoie à l'exportateur de cette offre après le succès du traitement d'une demande d'exportation.

NOTE 1 – L'agent T-DUA peut choisir les noms à l'aide d'un algorithme quelconque – pourvu que les identificateurs résultants soient uniques. Un compteur dont on incrémente la valeur suffit.

```
sOfferId ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString{ub-s-offer-id}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                         id-trader-at-sOfferId}
```

NOTE 2 – L'exportateur reçoit un identificateur d'offre de service qui est la représentation sous forme de chaîne ASN.1 du nom d'annuaire partiel (sous la racine du courtier) de l'entrée offre de service. Dans l'exemple de la Figure 3, {sOfferId, 5} pourrait être un identificateur d'offre de service renvoyé à l'exportateur.

### 6.4.2 serviceInterfaceId

Cet identificateur contient le nom d'une entrée interface (voir 6.7) qui comporte des informations sur l'interface du service exporté. Le courtier utilise ce nom pour lire l'entrée interface en vue de récupérer des informations (par exemple l'adresse de présentation ou la référence d'interface d'ingénierie) relatives à cette interface.

```
serviceInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-serviceInterfaceId}
```

NOTE – Lorsqu'une seule infrastructure est en jeu, on pourrait remplacer la spécification relative à serviceInterfaceId ci-dessus par:

```
ServiceIntId ::= SEQUENCE {
    serviceInterfaceType DirectoryString{ub-service-interface-type}
    interfaceRef         OPEN TYPE}

serviceInterfaceId ATTRIBUTE ::= {
    WITH SYNTAX          ServiceIntId
    EQUALITY MATCHING RULE caseExactMatch
    ID                  id-trader-at-serviceInterfaceId}
```

### 6.4.3 serviceTypeId

Il s'agit de l'identificateur du type correspondant au service proposé dans une offre. Cet identificateur contrôle le type d'interface et le type de propriétés de service qui peuvent être associés à cette offre de service.

```
serviceTypeId ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                  id-trader-at-serviceTypeId }
```

### 6.4.4 hasDynamicProperties

S'il a la valeur 'true' (vrai), cet attribut indique que cette offre de service contient des propriétés dynamiques, c'est-à-dire des propriétés dont il faut obtenir les valeurs lorsque c'est nécessaire pour une mise en concordance ou une description d'offres de service. Dans sa politique d'importation, un importateur peut demander de ne pas tenir compte des offres comportant des propriétés dynamiques.

```
hasDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX          BOOLEAN
    EQUALITY MATCHING RULE booleanMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-hasDynamicProperties}
```

#### 6.4.5 hasModifiableProperties

S'il a la valeur 'true' (vrai), cet attribut indique que cette offre de service contient des propriétés qui ne sont pas en lecture seule, c'est-à-dire des propriétés dont les valeurs peuvent être modifiées par l'exportateur lorsque c'est nécessaire. Dans sa politique d'importation, un importateur peut demander de ne pas tenir compte des offres comportant des propriétés modifiables.

```
hasModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                               BOOLEAN
    EQUALITY MATCHING RULE   booleanMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-hasModifiableProperties}
```

#### 6.4.6 dynamicProps

Cet attribut est positionné lorsqu'une offre de service contient des propriétés dynamiques, c'est-à-dire des propriétés dont il faut obtenir les valeurs lorsqu'elles sont nécessaires pour une mise en concordance ou une description d'offres de service. Chaque exportateur peut désigner les propriétés (obligatoires ou facultatives) qui sont dynamiques. Toutefois, pour pouvoir être désignée comme dynamique, une propriété ne doit pas être en lecture seule. L'attribut dynamicProps contient les informations qui sont nécessaires pour obtenir les valeurs d'une ou de plusieurs propriétés dynamiques. Une valeur dynamicPropValue est fixée pour chaque propriété dynamique d'une offre de service.

```
dynamicProps ATTRIBUTE ::= {
    WITH SYNTAX                               SEQUENCE OF DynamicPropValue
    ID                        id-trader-at-dynamicProps}

dynamicPropValue ::= SEQUENCE {
    propertyType                OBJECT IDENTIFIER,
    dynamicPropEvalIf           DistinguishedName,
    extraInfo                    DirectoryString{ub-dynamic-value-extra-info}}

DynamicPropValueMatch MATCHING-RULE ::= (
    SYNTAX                        DynamicPropValue,
    ID                            id-trader-mr-dynamicPropValueMatch}

-- La règle renvoie TRUE (vrai) si deux valeurs contiennent exactement les mêmes
-- caractères.
```

#### 6.4.7 Autres attributs X.500

Outre les attributs déjà énumérés, une entrée offre de service contient normalement d'autres attributs qui représentent les propriétés de service. Les attributs de propriété de service sont propres au type de service et sont contrôlés par une classe d'objets auxiliaire. Le type de propriétés de service spécifie si une propriété est obligatoire ou facultative et (avec l'information de contrôle d'accès) si un utilisateur peut en modifier la valeur. L'attribut dynamicProps est propre à une offre donnée, par exemple, l'offre d'un service particulier par un exportateur pourra ne comporter aucune valeur dynamique tandis qu'une offre de service différente du même type de service pourra contenir plusieurs propriétés dynamiques.

D'autres attributs X.500 requis par une réalisation ou une application donnée peuvent aussi être inclus. Donnons quelques exemples d'autres attributs X.500 pouvant figurer dans cette entrée:

- information de contrôle d'accès (sur l'entrée offre de service);
- date d'expiration d'une offre de service;
- information de contact humain pour le service, y compris une description textuelle du service.

## 6.5 Entrée lien du courtier

Les entrées lien du courtier (zéro, une ou plusieurs) sont situées immédiatement sous l'entrée courtier dans le sous-arbre d'information. Chaque entrée lien du courtier contient les renseignements relatifs à un lien du courtier et un seul. L'ensemble des entrées lien du courtier constituent l'ensemble des liens du courtier. Chaque entrée lien du courtier est dénommée au moyen de l'attribut linkId de cette entrée.

```
traderLinkEntryNF NAME-FORM ::= {
    NAMES                traderLinkEntry
    WITH ATTRIBUTES      {linkId}
    ID                   id-trader-nf-traderLink}

traderLinkEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {linkName | linkId | targetTraderInterfaceId |
                        defPassOnFollowRule | limitingFollowRule}
    ID                   id-trader-oc-traderLink}
```

### 6.5.1 linkName

Il s'agit du nom du lien tel qu'il est fourni par l'administrateur du courtier source au moment de la création du lien. Le nom doit être unique parmi tous les noms de lien détenus par le courtier.

```
linkName ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-link-name}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-linkName}
```

### 6.5.2 linkId

Il s'agit de l'identificateur du lien qui est attribué par le courtier. Il constitue le nom RDN de l'entrée lien du courtier. L'identificateur doit être unique parmi tous les identificateurs de lien détenus par ce courtier.

NOTE – L'agent T-DUA peut choisir les noms à l'aide d'un algorithme quelconque – pourvu que les identificateurs résultants soient uniques. Un compteur dont on incrémente la valeur suffit.

```
linkId ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-link-id}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-linkId}
```

### 6.5.3 targetTraderInterfaceId

Cet attribut contient le nom de l'interface du courtier cible pointé par le lien. Le nom est utilisé pour la consultation de l'adresse de présentation du courtier distant.

```
targetTraderInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                   id-trader-at-targetTraderInterfaceId}
```

### 6.5.4 defPassOnFollowRule

Le comportement de poursuite par défaut associé à un lien est spécifié au moment de la création du lien. Sa valeur ne doit pas dépasser celle du comportement limitingFollowRule associé à ce lien.

```
defPassOnFollowRule ATTRIBUTE ::= {
    WITH SYNTAX          FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-defPassOnFollowRule }
```

### 6.5.5 limitingFollowRule

Le comportement de poursuite limite associé à un lien donné est spécifié au moment de la création du lien. Sa valeur ne doit pas dépasser celle du comportement maxLinkFollowPolicy du courtier source au moment de la création du lien. Ceci est toutefois admissible lorsqu'une modification ultérieure du comportement maxLinkFollowPolicy du courtier a pour conséquence que la valeur de limitingFollowRule dépasse celle du comportement maxLinkFollowPolicy modifié du courtier.

```
limitingFollowRule ATTRIBUTE ::= {
    WITH SYNTAX          FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                   id-trader-at-limitingFollowRule }
```

### 6.5.6 Autres attributs X.500

Outre les attributs X.500 déjà énumérés, une entrée lien du courtier contient normalement d'autres attributs qui représentent des informations du courtier cible telles qu'elles sont perçues par le courtier source. Grâce à ces attributs de lien, le courtier source peut décider plus facilement du prochain lien à suivre. Ces attributs X.500 et d'autres attributs X.500 requis par une réalisation ou une application donnée peuvent être inclus sous la forme de classes d'objets auxiliaires. Donnons quelques exemples d'autres attributs X.500 pouvant figurer dans cette entrée:

- information de contrôle d'accès (sur l'entrée lien du courtier);
- attributs relatifs à l'entrée courtier (distant) lié, y compris une information de contact humain pour le courtier lié et une description textuelle du courtier lié;
- la période de validité de ce lien;
- la date de dernière invocation du lien.

## 6.6 Entrée offre de délégation

Les entrées offre de délégation (zéro, une ou plusieurs) sont situées immédiatement sous l'entrée courtier dans le sous-arbre d'information. Aucune entrée offre de délégation n'est nécessaire si l'attribut supportsProxyOffers de l'entrée courtier est mis à 'false' (faux). Si cet attribut est mis à 'true' (vrai), chaque entrée offre de délégation contient les renseignements relatifs à une offre de délégation et une seule. L'ensemble des entrées offre de délégation constituent l'ensemble des offres de délégation. Chaque entrée offre de délégation comporte une classe d'objets structurelle proxyOfferEntry et exactement une classe d'objets auxiliaire parmi celles qui sont énumérées dans la règle de contenu applicable à cette entrée. Chaque classe d'objets auxiliaire correspond à un type de propriétés de service et donne la définition des propriétés de service qui sont obligatoires et de celles qui sont facultatives. Chaque entrée offre de délégation contient donc les attributs obligatoires et facultatifs relatifs à cette entrée ainsi que les attributs obligatoires et facultatifs déterminés par la classe d'objets auxiliaire correspondante, qui spécifie les propriétés obligatoires et facultatives du service. Chaque entrée offre de délégation est dénommée au moyen de l'attribut proxyOfferId de l'entrée.

```

proxyOfferEntryNF NAME-FORM ::= {
    NAMES                proxyOfferEntry
    WITH ATTRIBUTES      {proxyOfferId}
    ID                    id-trader-nf-proxyOffer}

proxyOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN         {proxyOfferId | proxyLookUpInterfaceId |
                        hasDynamicProperties | hasModifiableProperties |
                        ifMatchAll | constraintRecipe}
    MAY CONTAIN          { dynamicProps }
    ID                    id-trader-oc-proxyOffer}

```

Une offre de délégation est la combinaison d'une offre de service avec une forme de lien restreint. Comme les offres de service normales, les offres de délégation possèdent un type de service et un ensemble de propriétés de service. Toutefois, une offre de délégation n'inclut pas d'interface à laquelle le service offert est fourni. En revanche, elle fournit une interface de consultation, à partir de laquelle on invoque une opération d'interrogation modifiée et à partir de laquelle on peut déterminer en cours d'exécution l'interface à laquelle le service sera fourni.

En outre, une offre de délégation comporte un attribut constraintRecipe (utilisé pour formuler une contrainte modifiée dans l'interrogation modifiée). Une offre de délégation peut aussi comporter une séquence de couples nom-valeur à annexer à la politique d'origine de l'importateur pour former la politique de l'importateur dans l'interrogation modifiée.

### 6.6.1 proxyOfferId

Il s'agit du nom RDN de l'entrée et il doit être unique parmi tous les noms d'offre de délégation détenus par le courtier. L'identificateur proxyOfferId fait partie de l'identificateur d'offre de délégation que l'agent T-DUA renvoie à l'exportateur de cette offre de délégation après le succès du traitement d'une demande d'exportation de cette offre.

NOTE 1 – L'agent T-DUA peut choisir les noms à l'aide d'un algorithme quelconque – pourvu que les noms résultants soient uniques. Un compteur dont on incrémente la valeur suffit.

NOTE 2 – L'exportateur reçoit un identificateur d'offre de délégation qui est la représentation sous forme de chaîne ASN.1 du nom d'annuaire partiel (sous la racine du courtier) de l'entrée offre de délégation.

```

proxyOfferId ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-proxy-offer-id}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                    id-trader-at-proxyOfferId}

```

### 6.6.2 proxyLookUpInterfaceId

Il s'agit du nom d'une entrée représentant une interface à laquelle une opération d'interrogation de courtier peut être acceptée. Le courtier utilise ce nom pour lire l'entrée en vue de récupérer des informations (par exemple l'adresse de présentation) relatives à cette interface.

```
proxyLookUpInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-proxyLookUpInterfaceId}
```

### 6.6.3 constraintRecipe

Cet attribut donne la recette permettant de convertir une contrainte de concordance de l'importateur figurant dans la demande d'interrogation d'origine en une nouvelle contrainte de concordance à utiliser dans la demande d'interrogation modifiée au niveau de l'interface de consultation associée à l'offre de délégation. Le langage relatif aux contraintes est spécifié dans l'Annexe C de la Rec. UIT-T X.950 | ISO/CEI 13235-1.

```
constraintRecipe ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-constraint-recipe}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-constraintRecipe}
```

### 6.6.4 ifMatchAll

Si cet attribut a la valeur 'true' (vrai), le courtier doit mettre en concordance cette offre de délégation pour une interrogation d'importateur uniquement avec le type de service; autrement dit, il n'est pas nécessaire de mettre en concordance l'expression de "contrainte" de l'importateur avec les propriétés associées à cette offre de délégation pour une recherche.

```
ifMatchAll ATTRIBUTE ::= {
    WITH SYNTAX          BOOLEAN
    EQUALITY MATCHING RULE booleanMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-ifMatchAll }
```

### 6.6.5 Autres attributs X.500

Outre les attributs déjà énumérés, une entrée offre de délégation – comme une entrée offre de service – contient normalement d'autres attributs qui représentent les propriétés de service et indique s'il existe des propriétés dynamiques et des propriétés modifiables. Les attributs de propriété de service sont propres au type de service et sont contrôlés par une classe d'objets auxiliaire. Un type d'attribut de propriété spécifie si un attribut est obligatoire ou facultatif et (avec l'information de contrôle d'accès) si un utilisateur peut en modifier la valeur. S'il existe des propriétés dynamiques, elles sont stockées dans l'attribut dynamicProps (voir 6.4.6).

La spécification de la fonction de courtage donnée dans la Rec. UIT-T X.950 | ISO/CEI 13235-1 permet aussi d'inclure d'autres couples nom-valeur [paramètre (policies\_to\_pass\_on) politiques à transmettre] dans une offre de délégation. Il faut, le cas échéant, annexer ces couples nom-valeur au paramètre "policy" (politique) de la demande d'interrogation d'origine, afin de les transmettre à un atelier de service qui les utilisera pendant la création de service. Ces couples nom-valeur sont représentés sous la forme de couples type d'attribut et valeur d'attribut X.500.

D'autres attributs X.500 requis par une réalisation ou une application donnée peuvent être inclus. Donnons quelques exemples d'autres attributs X.500 pouvant figurer dans cette entrée:

- information de contrôle d'accès (sur l'entrée offre de délégation);
- information de contact humain pour l'offre de délégation, y compris une description textuelle du service.

## 6.7 Autres entrées X.500 utilisées par l'agent T-DUA

En plus des cinq types d'entrée figurant dans l'objet informationnel courtier, l'agent T-DUA peut aussi utiliser d'autres entrées de la base de données X.500. Citons par exemple des entrées représentant:

- des individus, des unités organisationnelles ou des organisations, qui peuvent stocker des informations d'authentification;
- des interfaces, qui contiennent des informations relatives à une interface, par exemple l'adresse (ou une référence d'interface d'ingénierie) utilisée pour accéder à cette interface. La Rec. UIT-T X.950 | ISO/CEI 13235-1 relative à la fonction de courtage indique qu'il faut spécifier une référence d'interface d'ingénierie pour chaque infrastructure. Pour conserver une souplesse maximale, chaque interface de la spécification de traitement est représentée par une entrée d'annuaire distincte. L'entrée d'annuaire peut posséder des attributs adaptés à chaque infrastructure, le cas échéant. Un certain niveau d'adressage indirect est donc ajouté pour l'accès aux adresses d'interface; le courtier stocke le nom d'une entrée contenant les informations d'interface et c'est cette entrée à laquelle on accède pour récupérer l'adresse. L'avantage est qu'en cas de changement de l'adresse utilisée pour accéder à l'interface, il n'est nécessaire de modifier qu'une seule entrée, et non toutes les références à cette interface pour tous les courtiers. Ceci est particulièrement important lorsque de nombreux courtiers doivent connaître l'interface – cas d'un répertoire de types par exemple.

Les entrées interface contiennent un attribut `interfaceReference` [référence d'interface] et un attribut `interfaceType` [type d'interface]. L'identificateur d'objet associé au type d'interface doit être connu du répertoire de types.

```
interfaceEntry OBJECT-CLASS ::= {
    SUBCLASS OF                {top}
    KIND                        auxiliary
    MUST CONTAIN                {interfaceReference | interfaceType}
    ID                          id-trader-oc-interfaceEntry}

interfaceReference ATTRIBUTE ::= {
    WITH SYNTAX                 InterfaceId
    EQUALITY MATCHING RULE     caseExactMatch
    ID                          id-trader-at-interfaceReference}

InterfaceId ::=                DirectoryString {ub-interface-id}

interfaceType ATTRIBUTE ::= {
    WITH SYNTAX                 InterfaceTypeName
    EQUALITY MATCHING RULE     objectIdentifierMatch
    ID                          id-trader-at-interfaceType}
InterfaceTypeName ::=         OBJECT IDENTIFIER
```

## 7 Opérations

Le présent article contient la description du mappage des fonctions de courtage spécifiées dans la Rec. UIT-T X.950 | ISO/CEI 13235-1 et des opérations que l'agent T-DUA exécute sur l'annuaire X.500 pour réaliser ces fonctions. On suppose que les exportateurs et les importateurs communiquent avec l'agent T-DUA via un protocole de courtage qui prend en charge la fonctionnalité requise dans la Rec. UIT-T X.950 | ISO/CEI 13235-1.

## 7.1 Initialisation

Lorsqu'un nouveau courtier est établi, l'administrateur crée l'entrée courtier (soit directement soit par l'intermédiaire d'un programme qui lit des informations de configuration). La nouvelle entrée contient les informations de configuration relatives au nouveau courtier.

A chaque démarrage du courtier (agent T-DUA), celui-ci se rattache à l'annuaire et lit l'entrée courtier pour découvrir les informations le concernant. Les seules informations d'amorçage requises par l'agent T-DUA sont les suivantes:

- le nom distinctif de l'entrée associée à son courtier;
- un mot de passe.

Une fois le courtier créé, l'agent T-DUA se rattache à l'agent DSA X.500 et reste rattaché jusqu'à la destruction du courtier (sauf pour les pannes du côté du courtier ou du côté du système X.500). Le courtier doit se rattacher à l'agent DSA avec suffisamment de privilèges pour pouvoir modifier l'entrée courtier et pouvoir créer, lire, supprimer et modifier des entrées offre de service et des entrées lien du courtier et, en cas de prise en charge d'offres de délégation, pouvoir créer, lire et supprimer des entrées offre de délégation.

## 7.2 Opérations du client

Un client utilise un protocole de courtage pour communiquer avec l'agent T-DUA. Ce protocole n'est pas défini dans la présente Spécification. L'opération demandée par le client se traduit par une ou plusieurs opérations sur la base de données X.500. Le mappage des opérations du courtier sur des opérations X.500 est décrit dans les paragraphes qui suivent. Un mappage a lieu uniquement après que l'agent T-DUA a vérifié la syntaxe de chaque paramètre telle qu'elle est spécifiée dans l'Annexe A de la Rec. UIT-T X.950 | ISO/CEI 13235-1; un paramètre dont la syntaxe est incorrecte est renvoyé sous forme d'anomalie avec le préfixe 'illegal' (illégal) et l'exécution de l'opération n'est pas poursuivie.

Plusieurs opérations de courtier nécessitent une certaine forme de sécurité (par exemple seul un client jouant le rôle d'un administrateur peut ajouter ou supprimer des liens). Cette sécurité s'appuie sur la sécurité fournie par la X.500. Pour exécuter une opération de courtier, un client doit fournir un nom et un mot de passe. Dans la présente Spécification, ce nom et ce mot de passe sont supposés être, ou se projeter directement en, un nom et un mot de passe d'utilisateur X.500, éventuellement protégés d'une certaine manière. La façon dont cela est réalisé dépend du protocole de courtage utilisé pour rattacher un client à un courtier.

L'agent T-DUA utilise cette authentification pour se rattacher à l'agent DSA X.500 en tant qu'utilisateur. La X.500 permet de protéger le mot de passe lorsque celui-ci est envoyé dans le réseau entre l'agent T-DUA et l'agent DSA X.500. Cette approche a une incidence au niveau de la sécurité: le client doit faire confiance à l'agent T-DUA, étant donné que ce dernier peut voir le mot de passe du client. Le protocole X.500 donne des mécanismes de protection des mots de passe lorsque ceux-ci sont transmis entre agents DUA et agents DSA. Il devrait être possible d'étendre cette protection entre un client et l'agent DSA car l'agent T-DUA n'a pas besoin de connaître le mot de passe.

Une fois rattaché à l'agent DSA en tant que client, l'agent T-DUA utilise les caractéristiques de sécurité X.500 pour contrôler l'accès aux entrées de l'arbre d'information du courtier. Aucune anomalie spécifique relative au contrôle d'accès n'est définie dans la Rec. UIT-T X.950 | ISO/CEI 13235-1 et une anomalie système est renvoyée lorsque l'accès n'est pas autorisé (par exemple, dans l'architecture CORBA, l'anomalie normalisée No\_Permission [pas d'autorisation] est signalée).

## 7.3 Opérations de registre

Les opérations de registre forment un ensemble d'opérations qui permettent à un client d'exporter, de retirer, de modifier et de décrire des offres de service d'un courtier. En outre, il existe une opération permettant de résoudre un nom de courtier par rapport à un contexte.

### 7.3.1 Exportation

Le courtier reçoit une opération d'exportation. Cette opération est mappée sur une opération d'adjonction d'entrée X.500 pour ajouter une nouvelle entrée offre de service à l'annuaire X.500. Les données de l'opération d'exportation sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
reference (référence)	Nom	Il s'agit du nom de l'entrée de la base de données X.500 qui contient l'adresse de l'interface de service. Voir 6.7.
type	Identificateur d'objet	Il s'agit du type de service. On suppose que ce paramètre correspond à l'identificateur du type de service. Dans le courtier X.500, il sert à obtenir la classe d'objets auxiliaire correspondant à la nouvelle entrée.
propriétés (propriétés)	Quelconque (facultatif)	Les véritables attributs X.500 fournis par l'exportateur pour les propriétés de service sont contrôlés par l'identificateur de type de propriétés de service.

L'agent T-DUA mappe l'opération d'exportation sur une opération d'adjonction d'entrée X.500. Le nom de la nouvelle entrée est choisi par l'agent T-DUA de telle sorte qu'il soit unique parmi toutes les offres de service. On peut utiliser n'importe quel schéma; un compteur dont on incrémente la valeur suffit. La classe d'objets structurelle de la nouvelle entrée est `serviceOffer`. La classe d'objets auxiliaire correspond à l'identificateur de type de propriétés de service (identificateur d'objet) (OID) associé à l'identificateur de type de service spécifié par l'exportateur. L'agent T-DUA utilise son répertoire de types pour obtenir l'identificateur de type d'interface et l'identificateur de type de propriétés de service associés au type de service.

Une information de contrôle d'accès doit être insérée dans la nouvelle entrée afin d'empêcher à d'autres utilisateurs X.500 d'accéder aux informations du courtier ou de les modifier. Une autre information de contrôle d'accès est ajoutée afin de permettre à l'agent DSA X.500 de contrôler l'accès aux informations du courtier par les clients. Ceci peut s'étendre de façon à n'accorder qu'à certains clients le privilège d'exporter des services.

Le paramètre "reference" donne le nom de l'interface à laquelle le service est fourni. L'entrée correspondante est lue en vue d'obtenir le type qui est associé à cette interface. Si l'on peut déterminer que ce type n'est pas un sous-type du type d'interface associé au type de service spécifié, l'anomalie `InterfaceTypeMismatch` (non-concordance de type d'interface) peut être renvoyée. L'anomalie `InvalidInterfaceRef` (référence d'interface non valide) peut être signalée si le paramètre contient 'nil' (néant).

Si l'identificateur de type de service ou la classe d'objets auxiliaire n'est pas connu du courtier, l'agent DSA renvoie une erreur de mise à jour de type `objectClassViolation` (transgression de classe d'objets). L'agent T-DUA renvoie l'anomalie `UnknownServiceType` (type de service inconnu). L'identificateur de type de service est stocké dans l'entrée offre de service.

Les valeurs de propriétés de service qui peuvent/doivent être présentes dans la nouvelle entrée sont contrôlées par la classe d'objets auxiliaire (identificateur de type de propriétés de service) de l'entrée. Si l'exportateur ne fournit pas les attributs obligatoires nécessaires pour l'identificateur de type de propriétés de service spécifié, l'agent DSA renvoie une erreur de mise à jour de type `objectClassViolation`. L'agent T-DUA renvoie l'anomalie `MissingMandatoryProperty` (propriété obligatoire manquante). De même, si l'utilisateur fournit des propriétés de service qui ne figurent pas dans la définition du service, une erreur de mise à jour de type `objectClassViolation` est renvoyée. (La spécification actuelle de la Rec. UIT-T X.950 | ISO/CEI 13235-1 permet à l'exportateur d'insérer une propriété facultative occasionnelle qui ne figure pas dans le type de propriétés de service. Toutefois, ceci n'est pas autorisé dans la X.500. L'exportateur doit d'abord spécifier un sous-type qui comprenne la propriété supplémentaire avant l'exportation.) Dans la vérification de schéma de la X.500, on ne vérifie pas si les propriétés de service fournies par l'exportateur sont valides (par exemple si le type d'imprimante correspond à l'une des trois valeurs valides). C'est l'agent T-DUA qui doit effectuer cette vérification et l'anomalie `PropertyTypeMismatch` (non-concordance de type de propriété) est signalée si la valeur de propriété n'est pas du type requis. Si l'erreur d'attribut `undefinedAttributeType` (type d'attribut non défini) se produit, l'agent T-DUA peut directement la mapper sur l'anomalie `PropertyTypeMismatch`.

L'agent T-DUA vérifie également si des propriétés dynamiques sont présentes dans l'offre. Pour chaque propriété dynamique de service identifiée (voir l'article 9), l'agent T-DUA remplace sa valeur par une valeur fictive et crée une valeur `dynamicPropValue` (valeur de propriété dynamique) associée dans l'attribut `dynamicProps` de l'entrée. L'agent T-DUA met aussi l'attribut `hasDynamicProperties` de l'entrée offre à 'true' (vrai) si l'offre contient des valeurs dynamiques. Si une propriété à laquelle est associée une valeur dynamique est non modifiable conformément à l'information de contrôle d'accès relative à la propriété, l'anomalie `ReadOnlyDynamicProperty` (propriété dynamique en lecture seule) est renvoyée.

Si un courtier ne prend pas en charge de propriété dynamique [c'est-à-dire si l'attribut de courtier `supportsDynamicProperties` est mis à 'false' (faux)] et si l'offre contient des propriétés dynamiques, l'anomalie `PropertyTypeMismatch` (non-concordance de type de propriété) est signalée et l'offre de service n'est pas acceptée.

Si toutes les propriétés du service sont non modifiables, l'attribut `hasModifiableProperties` de l'entrée offre est mis à 'false' (faux).

Si au moins deux propriétés de même nom sont incluses dans la liste de propriétés, l'anomalie `DuplicatePropertyName` (nom de propriété double) est signalée.

Si l'exportation aboutit, l'agent T-DUA signale ce succès avec l'information suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
<code>offerId</code> (identificateur d'offre)	Chaîne d'annuaire	Identificateur d'offre de service

où `offerId` est la représentation sous forme de chaîne d'un nom d'annuaire partiel, qui donne le nom d'accès à la nouvelle offre de service depuis l'entrée courtier. Le nom RDN (`sOfferId`) de la nouvelle entrée offre de service constitue le dernier nom RDN de la séquence de noms.

### 7.3.2 Retrait

L'agent T-DUA reçoit une demande de suppression d'une offre de service. L'opération de retrait est mappée sur une opération de suppression d'entrée X.500 pour supprimer l'entrée offre de service. La donnée de l'opération de retrait est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
<code>id</code> (identificateur)	Chaîne d'annuaire	Identificateur d'offre de service

L'entrée est identifiée à l'aide du paramètre "id", qui donne le nom distinctif partiel identifiant de manière univoque l'offre de service. Cette opération supprime tout simplement l'entrée offre de service.

Si le client fournit un identificateur d'offre inconnu, l'opération de suppression d'entrée renvoie l'erreur de nom `noSuchObject` (objet inexistant). L'agent T-DUA utilise alors le nom distinctif partiel pour essayer de lire une entrée offre de délégation. Si la lecture aboutit, l'anomalie `NotServiceOfferId` (ce n'est pas un identificateur d'offre de service) est signalée. Si la lecture échoue, l'anomalie `UnknownOfferId` (identificateur d'offre inconnu) est signalée.

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de l'exportation de l'offre de service ainsi que sur la base de l'identificateur de client de l'opération de retrait. Si le client a des droits d'accès insuffisants, l'erreur de sécurité `insufficientAccessRights` (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'anomalie système. Si la réalisation nécessite que l'administrateur du courtier élimine les anciennes offres, l'administrateur doit aussi avoir l'autorisation de supprimer des offres.

### 7.3.3 Modification

Le courtier reçoit une demande de modification des valeurs des propriétés d'une offre de service. L'opération de modification est mappée sur une opération de lecture X.500 puis sur une opération de modification d'entrée X.500 pour modifier l'entrée offre de service. Les données de l'opération de modification sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
<code>id</code> (identificateur)	Chaîne d'annuaire	Identificateur d'offre de service
<code>del_list</code> (liste pour la suppression)	Ensemble d'identificateurs d'objet (facultatif)	Liste des propriétés de service qu'il faut supprimer de l'offre
<code>modify_list</code> (liste pour la modification)	Quelconque (facultatif)	Liste des propriétés de service pour lesquelles il faut ajouter ou remplacer des valeurs

L'agent T-DUA commence par lire l'entrée offre de service identifiée grâce à l'identificateur d'offre de service. Si cette entrée n'existe pas, l'agent DSA renvoie l'erreur de nom `noSuchObject` (objet inexistant), qui est projetée sur l'anomalie de courtier `UnknownOfferId` (identificateur d'offre inconnu). Si le client n'a pas l'autorisation voulue pour lire l'entrée, l'erreur de sécurité `insufficientAccessRights` (droits d'accès insuffisants) est renvoyée à l'agent T-DUA. Cette opération

de lecture X.500 est exécutée de sorte que l'agent T-DUA puisse dresser une liste des attributs de propriétés réellement présents dans l'entrée offre de service. L'agent T-DUA modifie l'entrée en conséquence: les propriétés figurant sur la liste pour la suppression sont supprimées, des valeurs de propriétés existantes sont modifiées et de nouvelles valeurs de propriétés sont ajoutées.

Une opération de modification d'entrée X.500 est alors appliquée, elle consiste à supprimer toutes les propriétés de service existantes puis à ajouter les propriétés de l'offre modifiée. Les paramètres relatifs à cette nouvelle offre de service ont les mêmes nom et fonction que dans le cas de l'opération d'exportation. Si un nom de la liste pour la suppression n'existe pas dans l'offre d'origine, l'anomalie UnknownPropertyName (nom de propriété inconnu) est renvoyée.

Il est possible que la nouvelle entrée transgresse le schéma (tel qu'il est représenté par la classe d'objets auxiliaire – l'identificateur de type de propriétés de service). Dans ce cas, lorsque l'opération de modification d'entrée X.500 est exécutée, l'agent DSA renvoie l'erreur de mise à jour objectClassViolation (transgression de classe d'objets) et l'offre de service existante reste inchangée. L'agent T-DUA mappe l'erreur de mise à jour sur diverses anomalies. L'anomalie MandatoryProperty (propriété obligatoire) est renvoyée si une propriété obligatoire est supprimée mais pas rajoutée. Un contrôle d'accès est appliqué par l'agent DSA si l'opération essaie de modifier la valeur d'une propriété non modifiable ou de supprimer une propriété non modifiable et l'anomalie ReadOnly (lecture seule) est renvoyée; l'anomalie PropertyTypeMismatch (non-concordance de type de propriété) est renvoyée si les valeurs ne correspondent pas aux types spécifiés.

Si la modification se traduit par le passage d'une propriété non dynamique en propriété dynamique, l'agent T-DUA doit ajouter la valeur dynamicPropValue (valeur de propriété dynamique) appropriée à l'attribut dynamicProps de l'entrée. Si la modification se traduit par le passage d'une propriété dynamique en propriété non dynamique, l'agent T-DUA doit supprimer la valeur fictive et la valeur dynamicPropValue associée dans l'attribut dynamicProps et ajouter la nouvelle valeur statique. Il peut aussi être nécessaire de repositionner le fanion hasDynamicProperties en conséquence.

Les résultats de l'adjonction de nouvelles propriétés sont équivalents à une exportation de l'offre de service et la discussion du 7.3.1 sur les erreurs qui peuvent se produire s'applique aussi ici.

L'invocation de l'opération de modification d'entrée conduit l'agent T-DUA à renvoyer l'anomalie NotImplemented (non mis en œuvre) si l'attribut de courtier supportsModifiableProperties a la valeur 'false' (faux).

### 7.3.4 Description

L'agent T-DUA reçoit une demande de description d'une offre de service. L'opération de description est mappée sur une opération de lecture X.500 pour lire l'entrée offre de service. La donnée de l'opération de description est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
id (identificateur)	Chaîne d'annuaire	Identificateur d'offre de service

Pour l'opération de lecture, l'entrée est identifiée à l'aide du paramètre "id" (nom distinctif partiel d'une entrée offre de service), 'allUserAttributes' (tous les attributs d'utilisateur) étant choisi comme information d'entrée. L'agent T-DUA doit examiner l'attribut dynamicProps de l'entrée renvoyée. Pour chaque valeur dynamicPropValue de l'attribut dynamicProps, la valeur fictive associée doit être supprimée et remplacée par la valeur renvoyée par l'opération d'évaluation des propriétés dynamiques invoquée à l'interface dynamicPropEvalIf (interface pour l'évaluation des propriétés dynamiques) associée pour récupérer la valeur courante de cette propriété (voir 7.9).

L'agent T-DUA filtre les attributs renvoyés et retransmet à son client les informations suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
reference (référence)	Nom	Nom de l'interface à laquelle le service est fourni
type	Identificateur d'objet	Identificateur de type de service de l'offre
properties (propriétés)	Ensemble d'attributs	Chaque type d'attribut mappe un nom de propriété et la valeur d'attribut associée mappe une valeur de propriété

Si le client fournit un identificateur d'offre de service inconnu, l'erreur de nom noSuchObject (objet inexistant) est renvoyée. L'agent T-DUA renvoie l'anomalie UnknownOfferId (identificateur d'offre inconnu).

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de l'exportation de l'offre de service ainsi que sur la base de l'identificateur de client de l'opération de description. Si le client a des droits d'accès insuffisants, l'erreur de sécurité `insufficientAccessRights` (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'erreur système appropriée.

### 7.3.5 Retrait avec contrainte

L'agent T-DUA reçoit une demande de suppression d'un ensemble d'offres de service d'un courtier. Cet ensemble est identifié de la même façon que pour une opération d'interrogation (voir 7.4.1) lorsque celle-ci identifie un ensemble d'offres à renvoyer à un importateur.

Les données de l'opération de retrait avec contrainte sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
type	Identificateur d'objet	Identificateur de type de service à utiliser pour choisir les offres qui doivent être supprimées
constr (contrainte)	Filtre	Contrainte utilisée pour choisir les offres qui doivent être supprimées

La première étape de l'opération consiste à construire un filtre X.500 à partir de la contrainte. La présence de propriétés dynamiques dans les offres de service complique cette étape. Chaque test de propriété de la contrainte est étendu de sorte qu'il aboutisse lorsque la valeur `dynamicPropValue` associée est présente. Par exemple, le test `'dotsPerInch = 600'` est étendu à `'dotsPerInch = 600 OR dotsPerInchDynamicPropValue present'`. Le filtre doit intégrer la concordance du type de service.

Une fois le filtre X.500 construit, l'agent T-DUA invoque une recherche X.500 pour trouver les entrées offre de service concordantes. On choisit `sOfferId` comme information d'entrée pour la recherche X.500. L'identificateur `sOfferId` correspond au nom RDN des offres.

Aucune erreur X.500 n'est signalée si une propriété de service figurant dans la contrainte spécifiée par le client n'est pas reconnue par l'agent DSA. De même, aucune erreur n'est signalée si un nom de propriété de service n'est pas reconnu par l'agent DSA.

L'opération de suppression d'entrée X.500 est utilisée pour supprimer chacune des offres de service concordantes trouvées par l'agent DSA. Cette opération supprime tout simplement l'entrée offre de service. L'entrée est identifiée au moyen de l'identificateur `sOfferId` qui est renvoyé par l'agent DSA pour les offres concordantes.

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de l'exportation de l'offre de service ainsi que sur la base de l'identificateur de client de l'opération de retrait. Si le client a des droits d'accès insuffisants, l'erreur de sécurité `insufficientAccessRights` (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'anomalie système. Si la réalisation nécessite que l'administrateur du courtier élimine les anciennes offres, l'administrateur doit aussi avoir l'autorisation de supprimer des offres.

L'anomalie `NoMatchingOffers` (pas d'offre concordante) est renvoyée si l'opération de recherche X.500 n'aboutit à aucune offre concordante.

### 7.3.6 Résolution

L'agent T-DUA reçoit une demande de résolution du nom d'un autre courtier par rapport à un contexte. Cette opération est notamment utilisée pour une exportation vers un courtier qui est connu par son nom (une séquence de noms de lien qui identifie un chemin de courtiers liés dans un graphe de courtage) et non par une référence d'interface. Le client fournit le nom d'une séquence de composantes de nom. Chaque composante est utilisée pour trouver un nom de lien concordant dans un graphe de courtage. La donnée de l'opération de résolution est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
name (nom)	Séquence de noms	Nom de courtier constitué par une séquence de noms, où chaque nom est un nom de lien

L'agent T-DUA extrait et supprime la première composante de la séquence (un nom de lien) et utilise cette composante comme un filtre dans une opération de recherche à un niveau sur ses entrées lien. On choisit l'identificateur `targetTraderInterfaceId` comme information d'entrée pour la recherche X.500.

Si la séquence comporte un autre nom, l'agent T-DUA utilise l'opération de lecture X.500 pour récupérer une adresse du courtier cible correspondant à l'identificateur `targetTraderInterfaceId` trouvé. L'agent T-DUA se connecte alors au courtier cible lié et invoque l'opération de résolution avec comme paramètre les noms résiduels. Ce processus se poursuit jusqu'à ce que la séquence soit vide; à ce stade, l'agent T-DUA (qui exécute la dernière recherche) renvoie le dernier identificateur `targetTraderInterfaceId` à son client, qui à son tour renvoie l'adresse d'interface à son client, et ainsi de suite. Eventuellement, la dernière adresse d'interface de courtier correspondant au dernier nom de courtier est renvoyée à l'invocateur d'origine de cette opération.

L'anomalie `UnknownTraderName` (nom de courtier inconnu) est renvoyée si l'une quelconque des opérations de recherche X.500 n'aboutit pas le long du chemin.

Si le client n'a pas l'autorisation voulue pour exécuter une opération de recherche sur les entrées lien, l'erreur de sécurité `insufficientAccessRights` (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'anomalie système.

## 7.4 Opérations de consultation

Les opérations de consultation permettent de réaliser la fonctionnalité d'importation de la fonction de courtage. L'interrogation est la seule opération. Elle sert à importer des entrées offre de service stockées dans un courtier ou dans un groupe de courtiers fédérés.

### 7.4.1 Opération d'interrogation

Le courtier reçoit une demande de recherche d'offres de service correspondant au type de service spécifié du client et restreintes à un cadre de recherche attendu identifié par la politique de l'importateur. Facultativement, l'ensemble des offres concordantes peut ensuite être filtré selon des critères de préférence du client. L'opération d'interrogation peut aussi incorporer les noms des propriétés dont les valeurs sont intéressantes pour l'importateur. Les données de l'opération d'interrogation sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
type	Identificateur d'objet	Il s'agit du type de service auquel les offres de service doivent correspondre
constr (contrainte)	Filtre	Ce paramètre représente la contrainte utilisée pour les critères de concordance des offres – voir ci-dessous
pref (préférence)	Non utilisé	Critères de préférence – utilisés uniquement par l'agent T-DUA
policies (politiques)	(Facultatif)	Attentes de l'importateur – voir ci-dessous
desired_props (propriétés souhaitées)	(Facultatif)	Propriétés de service intéressantes – voir ci-dessous
how_many (nombre)	Non utilisé	Contrôle de l'itérateur – utilisé uniquement par l'agent T-DUA

Compte tenu de la complexité d'une importation, la description est scindée en deux composantes: la recherche dans l'espace d'offres de service du courtier local et la recherche dans les espaces d'offres des courtiers fédérés. Ces deux composantes de recherche peuvent être effectuées séquentiellement ou en parallèle, ceci étant laissé au choix de l'agent T-DUA.

### 7.4.2 Politiques

La première étape de l'exécution d'une opération d'interrogation consiste à traiter les politiques. Les politiques sont représentées par des couples <nom, valeur>. Un importateur peut spécifier zéro, une ou plusieurs politiques propres qui indiquent ses attentes pour la recherche. Chaque courtier a ses propres politiques de recherche qui guident son comportement au cours de la recherche. Les politiques du courtier interagissent avec celles de l'importateur. Si une politique d'importateur n'est pas spécifiée, le courtier utilise la politique par défaut du courtier. Si la valeur d'une politique d'importateur dépasse la valeur maximale (ou limite) de politique fixée par le courtier, celui-ci remplace la valeur attendue par l'importateur par la valeur maximale/limite de la politique du courtier.

Il appartient à l'agent T-DUA de vérifier et de fixer les véritables valeurs maximales concernant les politiques de cardinalité et la politique "hop\_count" (nombre de sauts). Il lui appartient aussi d'insérer un nom de politique dans le paramètre renvoyé "limits\_applied" (limites appliquées), s'il est mis fin prématurément à une recherche en raison d'une valeur maximale.

Les courtiers prennent en charge trois politiques de capacités normalisées. Si un courtier ne prend pas en charge une capacité, les importateurs doivent s'en accommoder; la non-utilisation d'une capacité demandée par un importateur est signalée par l'insertion du nom de la politique de capacité dans le paramètre "limits-applied". Si toutefois un courtier prend en charge une capacité et qu'un importateur ne souhaite pas l'utiliser, ce courtier doit respecter le choix de l'importateur. La capacité du courtier s'applique lorsque l'importateur ne spécifie pas de capacité attendue. Les importateurs peuvent choisir la valeur des trois capacités normalisées suivantes:

- use\_proxy\_offers (utilisation d'offres de délégation) – Si la valeur est 'false' (faux), les recherches de concordances ne couvrent pas les entrées offre de délégation;
- use\_modifiable\_properties (utilisation de propriétés modifiables) – Si la valeur est 'false' (faux), l'agent T-DUA ajoute la condition nécessaire 'hasModifiableProperties = false' dans le filtre de concordance;
- use\_dynamic\_properties (utilisation de propriétés dynamiques) – Si la valeur est 'false' (faux), l'agent T-DUA ajoute la condition nécessaire 'hasDynamicProperties = false' dans le filtre de concordance.

L'étape suivante consiste à traiter la politique de comportement relative aux liens pour déterminer si un interfonctionnement peut se produire. Cette étape est décrite au 7.4.4.

Si un importateur déclare une politique "starting\_trader" (courtier de départ), on recherche la première composante du nom de courtier en la comparant au nom figurant dans chaque entrée lien. Si aucune concordance n'est trouvée, l'anomalie InvalidPolicyValue (valeur de politique non valide) est renvoyée. Dans le cas contraire, l'agent T-DUA invoque l'opération d'interrogation sur le lien en question et transmet la politique "starting\_trader", la première composante ayant été supprimée (voir aussi le 7.3.6 sur l'opération de résolution).

### 7.4.3 Recherche locale

Pour la recherche locale, on commence par construire un filtre X.500 afin de sélectionner les entrées correspondant à l'identificateur de type de service en question. Outre les offres de service qui correspondent exactement à l'identificateur de type de service, le filtre laisse aussi passer les entrées dont l'identificateur de type de service est un affinement (sous-classe) du type spécifié (identificateur de type de service). Toutefois, seules les concordances exactes par rapport au type de service sont retenues si l'importateur a fixé la politique d'importateur "exact\_type\_match" (concordance exacte par rapport au type) à 'true' (vrai) dans l'opération d'interrogation. Si le courtier prend en charge des propriétés dynamiques et que l'importateur souhaite utiliser des offres avec propriétés dynamiques, l'attribut hasDynamicProperties est choisi comme information d'entrée pour l'opération de recherche X.500. Une recherche à un niveau est effectuée sur les entrées offre de service pour voir s'il existe d'éventuelles offres de service qui comportent le type de service requis.

Un deuxième filtre est construit à partir de la contrainte spécifiée pour la concordance. Cette construction peut être rendue complexe par la prise en charge et la présence de propriétés dynamiques dans les offres de service (connaissance obtenue dans la première recherche). Si le courtier prend en charge des propriétés dynamiques et que l'importateur souhaite utiliser des offres avec propriétés dynamiques, chaque test de propriété de la contrainte est étendu de sorte qu'il aboutisse lorsque la valeur dynamicPropValue associée est présente. Par exemple, le test 'dotsPerInch = 600' est étendu à 'dotsPerInch = 600 OR dotsPerInchDynamicPropValue present'.

Le deuxième filtre introduit une autre condition nécessaire par rapport au premier filtre. L'agent T-DUA invoque une recherche X.500 pour trouver les entrées offre de service concordantes. Tous les attributs d'utilisateur sont choisis comme informations d'entrée X.500; il appartient à l'agent T-DUA de filtrer les attributs renvoyés. Un utilisateur peut souhaiter obtenir la description des valeurs pour aucune, certaines ou la totalité des propriétés de service.

Aucune erreur X.500 n'est signalée si une propriété de service figurant dans la contrainte spécifiée par le client n'est pas reconnue par l'agent DSA. De même, aucune erreur n'est signalée si un nom de propriété de service n'est pas reconnu par l'agent DSA. Toutefois, une erreur d'attribut noSuchAttributeOrValue (attribut ou valeur inexistant) est renvoyée si aucun attribut n'est renvoyé.

Si un agent DSA obtient de nombreuses offres de service après application du filtre, il n'est pas nécessaire qu'il les renvoie toutes. Dans ce cas, le qualificateur de résultats partiels est positionné avec la valeur 'sizelimitExceeded' (taille limite dépassée). Il est possible que l'agent T-DUA doive alors relancer la recherche en utilisant l'option de résultats paginés de l'opération de recherche X.500.

Les entrées renvoyées dans le cadre des recherches X.500 peuvent contenir des propriétés dynamiques. L'agent T-DUA doit vérifier, pour chaque entrée, si l'attribut `dynamicProps` est présent. S'il est présent, l'attribut fictif associé pour chaque valeur `dynamicPropValue` de l'attribut `dynamicProps` est supprimé du résultat de la recherche et l'interface associée pour l'évaluation de la propriété est invoquée afin de récupérer la valeur courante de cette propriété dynamique (voir le paragraphe 7.9 relatif à l'évaluation des propriétés dynamiques). L'agent T-DUA doit alors évaluer la contrainte de l'interrogation pour déterminer si les offres de service satisfont réellement la contrainte.

#### 7.4.4 Recherche parmi des courtiers fédérés

L'agent T-DUA utilise la politique "hop\_count" (nombre de sauts) pour contrôler la profondeur des liens traversés pour une importation. Lorsque cette politique a la valeur nulle, on cesse de propager l'importation parmi les courtiers liés. La valeur de cette politique est décrémentée de un chaque fois qu'un lien est traversé pour la transmission d'une opération d'interrogation.

A chaque lien sont associées deux politiques de poursuite "def\_pass\_on\_follow\_rule" et "limiting\_follow\_rule" qui sont fixées au moment de la création du lien. Le courtier possède quant à lui trois politiques relatives aux liens:

- `maxLinkFollowPolicy` – Limite la valeur de la politique "limiting\_follow\_rule" autorisée au moment du premier établissement du lien;
- `maxFollowPolicy` – Limite le comportement de poursuite pour tout lien du courtier pour une opération d'interrogation donnée;
- `defFollowPolicy` – Fournit la règle de poursuite de lien par défaut si un importateur n'en a pas spécifiée dans son interrogation.

La politique de poursuite associée à un lien particulier est donc l'une des deux suivantes:

- `min (trader.max_follow_policy, link.limiting_follow_rule, query.link_follow_rule)`;
- `min (trader.max_follow_policy, link.limiting_follow_rule, trader.def_follow_policy)`.

Si des liens appropriés doivent être suivis, l'agent T-DUA utilise l'opération de lecture X.500 pour projeter les noms `targetTraderInterfaceId` (figurant dans les entrées lien du courtier) sur les adresses de chaque courtier cible. L'agent T-DUA se connecte alors aux courtiers sélectionnés et leur transmet l'opération d'interrogation en spécifiant comme politique "link\_follow\_rule" l'une des deux suivantes:

- `min (trader.max_follow_policy, link.limiting_follow_rule, query.link_follow_rule)`;
- `min (trader.max_follow_policy, link.def_pass_on_follow_rule)`.

En outre, si le courtier source qui lance une opération d'interrogation parmi des courtiers fédérés souhaite inclure un identificateur d'opération d'interrogation dans le paramètre "policy" (politique), il peut le faire. Les courtiers ne sont pas tenus de générer un tel identificateur d'opération mais sont tenus de transmettre ce paramètre en aval d'un lien.

#### 7.4.5 Recherche d'offres de délégation

Si la politique "use\_proxy\_offer" (utilisation d'offres de délégation) de l'importateur n'a pas la valeur 'false' (faux) et que l'attribut `supportsProxyOffer` du courtier a la valeur 'true' (vrai), les recherches X.500 du 7.4.3 sont répétées pour les entrées offre de délégation. Pour chaque offre de délégation concordante, l'agent T-DUA applique une opération d'interrogation modifiée en utilisant les informations stockées dans l'offre de délégation concordante. Cet agent utilise ensuite l'opération de lecture X.500 pour projeter le nom `proxyLookUpInterfaceId` (figurant dans l'entrée offre de délégation) sur une adresse qu'il utilise pour invoquer l'opération d'interrogation modifiée.

#### 7.4.6 Offres de service renvoyées

Une fois que le courtier a trouvé un ensemble d'offres de service, le paramètre "pref" et les diverses politiques de cardinalité sont utilisés en vue d'ordonner et de limiter les offres renvoyées à l'importateur. Il s'agit d'une fonction de l'agent T-DUA et cette fonction n'est pas examinée plus avant dans la présente Spécification. La mise en œuvre d'un itérateur d'offre relève aussi de l'agent T-DUA.

L'anomalie `UnknownType` (nom de type de service inconnu) ou `NoMatchingType` (aucune offre avec le type de service requis) est renvoyée si l'une de ces conditions est vérifiée dans l'ensemble de l'espace de recherche.

## 7.5 Opérations relatives aux liens

### 7.5.1 Adjonction de lien

Le courtier reçoit une opération d'adjonction de lien. Cette opération est mappée sur une opération d'adjonction d'entrée X.500 pour ajouter une nouvelle entrée lien du courtier à l'annuaire X.500. Les données de l'opération d'adjonction de lien sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
name (nom)	Chaîne d'annuaire	Nom du nouveau lien
target (cible)	Nom	Nom d'un autre courtier
def_pass_on_follow_rule (règle de poursuite par défaut)	Option de poursuite	Ne doit pas dépasser la valeur de "limiting_follow_rule"
limiting_follow_rule (règle de poursuite limite)	Option de poursuite	Ne doit pas dépasser la valeur de "max_link_follow_policy" du courtier

L'agent T-DUA vérifie d'abord les valeurs admissibles pour les paramètres. L'anomalie DefaultFollowTooPermissive (poursuite par défaut trop peu contraignante) est signalée si la valeur du paramètre "def\_pass\_on\_follow\_rule" dépasse celle du paramètre "limiting\_follow\_rule". L'anomalie LimitingFollowTooPermissive (poursuite limite trop peu contraignante) est signalée si la valeur du paramètre "limiting\_follow\_rule" dépasse celle de la politique "max\_link\_follow\_policy" du courtier. L'agent T-DUA doit d'abord lire l'entrée courtier pour déterminer sa politique "max\_link\_follow\_policy". Il s'assure aussi que le nouveau nom de lien est unique parmi l'ensemble des liens du courtier en recherchant s'il existe déjà une instance du nom. Si une concordance est trouvée, l'anomalie DuplicateLinkname (nom de lien double) est signalée.

L'agent T-DUA mappe l'opération d'adjonction de lien sur une opération d'adjonction d'entrée X.500. Il choisit, pour la nouvelle entrée, un identificateur qui soit unique parmi tous les liens du courtier. On peut utiliser n'importe quel schéma; un compteur dont on incrémente la valeur suffit. La classe d'objets structurelle de la nouvelle entrée est traderLink. La classe d'objets auxiliaire correspondante peut aussi spécifier d'autres propriétés de lien qui peuvent être intégrées à l'opération d'adjonction d'entrée.

Une information de contrôle d'accès doit être insérée dans la nouvelle entrée afin d'empêcher à d'autres courtiers ou à d'autres utilisateurs X.500 d'accéder aux informations de lien ou de les modifier.

Si l'adjonction de lien aboutit, aucune anomalie n'est signalée.

### 7.5.2 Suppression de lien

L'agent T-DUA reçoit une demande de suppression d'un lien du courtier. La donnée de l'opération de suppression de lien est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
name (nom)	Chaîne d'annuaire	Identifie le lien à supprimer par le nom de lien

L'agent T-DUA doit d'abord appliquer l'opération de recherche X.500 aux entrées lien afin de rechercher une entrée possédant le nom de lien en question. Si la recherche n'aboutit pas, l'anomalie UnknownLinkname (nom de lien inconnu) est signalée.

L'agent T-DUA utilise alors l'opération de suppression d'entrée X.500 pour supprimer l'entrée lien du courtier en employant l'identificateur de lien de l'entrée concordante. Cette opération supprime tout simplement l'entrée lien du courtier.

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de la création du lien de courtier ainsi que sur la base de l'identificateur de client de l'opération de suppression. Si le client a des droits d'accès insuffisants, une erreur de sécurité insufficientAccessRights (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'anomalie système.

### 7.5.3 Modification de lien

Le courtier reçoit une demande de modification d'un lien du courtier. L'opération de modification est projetée sur une opération de recherche X.500 pour trouver le lien utilisant le nom de lien cherché puis sur une opération de modification d'entrée X.500 pour modifier l'entrée lien du courtier. Les données de l'opération de modification de lien sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
name (nom)	Chaîne d'annuaire	Identifie le lien à modifier
def_pass_on_follow_rule (règle de poursuite par défaut)	Option de poursuite	Ne doit pas dépasser la valeur de "limiting_follow_rule"
limiting_follow_rule (règle de poursuite limite)	Option de poursuite	Ne doit pas dépasser la valeur de "limiting_follow_policy" du courtier

L'agent T-DUA vérifie d'abord les valeurs admissibles pour les paramètres. L'anomalie DefaultFollowTooPermissive (poursuite par défaut trop peu contraignante) est signalée si la valeur du paramètre "def\_pass\_on\_follow\_rule" dépasse celle du paramètre "limiting\_follow\_rule". L'anomalie LimitingFollowTooPermissive (poursuite limite trop peu contraignante) est signalée si la valeur du paramètre "limiting\_follow\_rule" dépasse celle de la politique "max\_link\_follow\_policy" du courtier. L'agent T-DUA recherche ensuite l'entrée lien du courtier identifiée par le nom de lien. "allUserAttributes" (tous les attributs d'utilisateur) est choisi comme information d'entrée X.500. Si l'entrée recherchée n'existe pas, l'agent DSA renvoie l'erreur de nom noSuchObject (objet inexistant), qui est mappée sur l'anomalie UnknownLinkName (nom de lien inconnu).

Une opération de modification d'entrée X.500 est alors appliquée, elle consiste à supprimer la totalité du comportement de poursuite de lien existant (et les propriétés, le cas échéant) puis à ajouter le nouveau comportement de poursuite (et les propriétés, le cas échéant) à partir de l'opération de modification de lien. Si le client n'a pas l'autorisation voulue pour supprimer des attributs de l'entrée, une erreur de sécurité insufficientAccessRights (droits d'accès insuffisants) est renvoyée à l'agent T-DUA.

La classe d'objets auxiliaire associée au lien de courtier peut aussi spécifier des propriétés de lien qui peuvent être intégrées à l'opération de modification d'entrée. L'entrée nouvellement modifiée peut transgresser le schéma. Dans ce cas, lorsque l'entrée lien nouvellement modifiée est ajoutée, l'agent DSA renvoie l'erreur de mise à jour objectClassViolation (transgression de classe d'objets). Le lien du courtier existant reste inchangé. Dans le cadre de la vérification de schéma X.500, on ne peut pas vérifier que les valeurs sont celles qui sont spécifiées par le schéma de lien. Une erreur d'attribut undefinedAttributeType (type d'attribut non défini) se produit si l'agent DSA ne connaît pas l'un des attributs fournis par le client.

### 7.5.4 Description de lien

Le courtier reçoit une demande de renvoi de renseignements concernant un lien donné. Cette opération est mappée sur l'opération de recherche X.500. La donnée est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
name (nom)	Chaîne d'annuaire	Identifie le lien à récupérer

L'agent T-DUA recherche l'entrée lien du courtier identifiée par le nom de lien, 'allUserAttributes' (tous les attributs d'utilisateur) étant choisi comme information d'entrée. Si l'entrée recherchée n'existe pas, l'agent DSA renverra l'erreur de nom noSuchObject (objet inexistant), qui est mappée sur l'anomalie UnknownLinkName (nom de lien inconnu). Si le client n'a pas l'autorisation voulue pour rechercher l'entrée, une erreur de sécurité insufficientAccessRights (droits d'accès insuffisants) est renvoyée à l'agent T-DUA.

Une fois que les données ont été renvoyées par l'agent DSA X.500, l'agent T-DUA renvoie les données, à l'exception de l'identificateur linkId, au client.

### 7.5.5 Listage des liens

Le courtier reçoit une demande de renvoi des noms de tous les liens du courtier que le courtier connaît. Cette opération est mappée sur l'opération de recherche X.500. La donnée est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
(aucun)		

L'agent T-DUA exécute une opération de recherche X.500 pour renvoyer l'attribut linkName (nom de lien) de toutes les entrées lien du courtier. Il s'agit d'une recherche à un niveau sur les entrées lien. Les noms de lien (le cas échéant) sont renvoyés à l'utilisateur:

Paramètre	Syntaxe d'attribut X.500	Observation
LinkNameSeq (séquence de noms de lien)	Ensemble de chaînes d'annuaire	

où chaque chaîne d'annuaire (un nom de lien) est unique pour chaque entrée lien.

## 7.6 Opérations relatives aux offres de délégation

Les opérations relatives aux offres de délégation forment un ensemble d'opérations qui permettent à un client d'exporter, de retirer et de décrire des offres de délégation d'un courtier.

### 7.6.1 Exportation d'offre de délégation

Le courtier reçoit une opération d'exportation d'offre de délégation. Cette opération est mappée sur l'opération d'adjonction d'entrée X.500 pour ajouter une nouvelle entrée offre de délégation à l'annuaire X.500. Les données de l'opération d'exportation d'offre de délégation sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
type	Identificateur d'objet	Le nom de type est supposé être l'identificateur de type de service. Il sert à obtenir la classe d'objets auxiliaire correspondant à la nouvelle entrée.
target (cible)	Nom	Il s'agit du nom de l'entrée de la base de données X.500 qui contient l'adresse associée à l'identificateur proxyLookUpInterfaceId.
properties (propriétés)	Quelconque (facultatif)	Les véritables attributs X.500 fournis par l'exportateur pour les propriétés de service sont contrôlés par l'identificateur de type de propriétés de service.
if_match_all (cas de concordance totale)	Booléen	
recipe (recette)	Chaîne	
policies_to_pass_on (politiques à transmettre)	Ensemble d'attributs (facultatif)	

L'agent T-DUA mappe l'opération d'exportation d'offre de délégation sur l'opération d'adjonction d'entrée X.500. Le nom de la nouvelle entrée est choisi par l'agent T-DUA de telle sorte qu'il soit unique parmi toutes les offres de délégation. On peut utiliser n'importe quel schéma; un compteur dont on incrémente la valeur suffit. La classe d'objets structurelle de la nouvelle entrée est proxyOffer. La classe d'objets auxiliaire correspond à l'identificateur de type de propriétés de service (identificateur d'objet) associé à l'identificateur de type de service de l'offre de délégation. L'agent T-DUA utilise son répertoire de types pour obtenir l'identificateur de type d'interface et l'identificateur de type de propriétés de service associés à ce type de service.

Une information de contrôle d'accès doit être insérée dans la nouvelle entrée afin d'empêcher à d'autres utilisateurs X.500 d'accéder aux informations du courtier ou de les modifier. Une autre information de contrôle d'accès est ajoutée afin de permettre à l'agent DSA X.500 de contrôler l'accès aux informations du courtier par les clients. Ceci peut s'étendre de façon à n'accorder qu'à certains clients le privilège d'exporter des offres de délégation.

Si l'identificateur de type de service ou la classe d'objets auxiliaire n'est pas connu du courtier, l'agent DSA renvoie une erreur de mise à jour de type `objectClassViolation` (transgression de classe d'objets). L'agent T-DUA renvoie une anomalie `UnknownServiceType` (type de service inconnu). L'identificateur de type de service est stocké dans l'entrée offre de délégation.

Les valeurs de propriété de service qui peuvent/doivent être présentes dans la nouvelle entrée sont contrôlées par la classe d'objets auxiliaire (identificateur de type de propriétés de service) de l'entrée. Ces valeurs de propriété sont traitées de la même manière que dans le cas de l'opération d'exportation d'offre de service (voir 7.3.1).

Le paramètre "if\_match\_all" spécifie si cette offre nécessite une concordance par rapport au type de service uniquement.

Le paramètre "target" donne le nom de l'interface qui est utilisée pour obtenir l'interface du service au moment de l'exécution. Si ce paramètre a la valeur "nil" (néant), l'anomalie `InvalidInterfaceRef` (référence d'interface non valide) est signalée.

Le paramètre "recipe" indique la modification qui peut être appliquée à la contrainte d'une opération d'interrogation qui donne cette offre de délégation concordante du point de vue du type de service et indique les valeurs de propriété lorsque le paramètre "if\_match\_all" ne vaut pas 'true' (vrai).

Le paramètre "policies\_to\_pass\_on" est constitué de couples <nom, valeur> à annexer au paramètre "policy" (politique) de l'opération d'interrogation à venir afin de former le paramètre "policy" modifié de l'opération d'interrogation modifiée.

Si l'exportation de l'offre de délégation aboutit, l'agent T-DUA signale le succès avec l'information suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
OfferId (identificateur d'offre)	Chaîne d'annuaire	Identificateur d'offre de délégation

où OfferId est la représentation sous forme de chaîne du nom distinctif partiel de la nouvelle entrée offre de délégation.

### 7.6.2 Retrait d'offre de délégation

L'agent T-DUA reçoit une demande de suppression d'une offre de délégation. L'opération de retrait d'offre de délégation est mappée sur l'opération de suppression d'entrée X.500 pour supprimer l'entrée offre de délégation. La donnée de l'opération de retrait d'offre de délégation est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
id (identificateur)	Chaîne d'annuaire	Identificateur d'offre de délégation

Cette opération supprime tout simplement l'entrée offre de délégation. L'entrée est identifiée à l'aide du paramètre "id", qui identifie de manière univoque une entrée offre de délégation.

Si le client fournit un identificateur d'offre de délégation inconnu, l'opération de suppression d'entrée renvoie l'erreur de nom `noSuchObject` (objet inexistant). L'agent T-DUA utilise alors l'identificateur d'offre de délégation pour lire une entrée offre de service. Si la lecture aboutit, l'anomalie `NotProxyOfferId` (ce n'est pas un identificateur d'offre de délégation) est signalée. Si la lecture échoue, l'anomalie `UnknownOfferId` (identificateur d'offre inconnu) est signalée.

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de l'exportation de l'offre de délégation ainsi que sur la base de l'identificateur de client de l'opération de retrait d'offre de délégation. Si le client a des droits d'accès insuffisants, une erreur de sécurité `insufficientAccessRights` (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'anomalie système. Si la réalisation nécessite que l'administrateur du courtier élimine les anciennes offres de délégation, l'administrateur doit aussi avoir l'autorisation de supprimer des offres de délégation.

### 7.6.3 Description d'offre de délégation

L'agent T-DUA reçoit une demande de description d'une offre de délégation. L'opération de description est mappée sur l'opération de lecture X.500 pour lire l'entrée offre de délégation. La donnée de l'opération de description est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
id (identificateur)	Chaîne d'annuaire	Identificateur d'offre de délégation

L'entrée est identifiée à l'aide du paramètre "id", qui permet une identification univoque. 'allUserAttributes' (tous les attributs d'utilisateur) est choisi comme information d'entrée pour l'opération de lecture. Il n'est pas nécessaire que l'agent T-DUA évalue les valeurs des propriétés dynamiques de l'offre de délégation.

L'agent T-DUA filtre les attributs renvoyés et retransmet à son client les informations suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
type	Identificateur d'objet	Identificateur de type de service
target (cible)	Nom distinctif	Nom de l'entrée interface qui donne la référence de l'interface du service
serviceProperties (propriétés du service)	Ensemble d'attributs	Un type d'attribut mappe un nom de propriété et une valeur d'attribut mappe une valeur de propriété
if_no_match (cas d'aucune concordance)	Booléen	
recipe (recette)	Chaîne	
policies_to_pass_on (politiques à transmettre)	Ensemble d'attributs (le cas échéant)	

Si le client fournit un identificateur d'offre de délégation inconnu, l'opération de lecture renvoie l'erreur de nom noSuchObject (objet inexistant). L'agent T-DUA utilise alors l'identificateur d'offre de délégation pour essayer de lire une entrée offre de service. Si la lecture aboutit, l'anomalie NotProxyOfferId (ce n'est pas un identificateur d'offre de délégation) est signalée. Si la lecture échoue, l'anomalie UnknownOfferId (identificateur d'offre inconnu) est signalée.

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de l'exportation de l'offre de délégation ainsi que sur la base de l'identificateur de client de l'opération de description d'offre de délégation. Si le client a des droits d'accès insuffisants, l'erreur de sécurité insufficientAccessRights (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'erreur système appropriée.

## 7.7 Opérations relatives aux attributs du courtier

Les propriétés et politiques du courtier sont exprimées sous la forme d'attributs X.500 dans l'entrée courtier. Tous les clients enregistrés d'un courtier peuvent *obtenir* des informations sur les caractéristiques du courtier en utilisant l'opération de lecture X.500 sur l'entrée courtier pour les attributs relatifs aux caractéristiques. Tous les exportateurs enregistrés peuvent, en outre, avoir un droit d'accès en lecture pour les attributs suivants: supportsModifiableProperties, supportsDynamicProperties, supportsProxyOffers et typeReposIf. Tous les importateurs enregistrés peuvent, en outre, avoir un droit d'accès en lecture pour les attributs de courtier suivants: defSearchCard, maxSearchCard, defMatchCard, maxMatchCard, defReturnCard, maxReturnCard, maxList, defHopCount, maxHopCount, supportsModifiableProperties, supportsDynamicProperties, supportsProxyOffers, defFollowPolicy, maxFollowPolicy et typeReposIf.

L'administrateur du courtier peut *obtenir* et *donner une valeur* à tous les attributs de courtier, où l'opération *obtenir* est mappée sur l'opération de lecture X.500 et l'opération *donner une valeur* à est mappée sur l'opération de modification d'entrée X.500.

## 7.8 Opérations administratives

### 7.8.1 Listage des offres

Le courtier reçoit une demande de renvoi des identificateurs de toutes les offres de service que le courtier connaît. Cette opération est mappée sur l'opération de recherche X.500. La donnée est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
how_many (nombre)	Non utilisé	Ce paramètre est conservé par l'agent T-DUA dans sa mise en œuvre de l'itérateur d'identificateur

L'agent T-DUA exécute une opération de recherche X.500 pour renvoyer les identificateurs de toutes les entrées offre de service. Il s'agit d'une recherche à un niveau de toutes les entrées offre de service. Aucune information n'est demandée. Ces identificateurs (le cas échéant) sont renvoyés à l'utilisateur sous la forme suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
ids (identificateurs)	Ensemble de chaînes d'annuaire	

où chaque identificateur (représentation sous forme de chaîne d'un nom distinctif partiel) identifie de manière univoque une entrée offre de service.

L'agent T-DUA est responsable de la mise en œuvre de l'itérateur d'identificateur.

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de l'exportation de l'offre de service et seuls les administrateurs peuvent exécuter l'opération de listage des offres. Si un client a des droits d'accès insuffisants, l'erreur de sécurité insuffisantAccessRights (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'erreur système appropriée.

### 7.8.2 Listage des offres de délégation

Le courtier reçoit une demande de renvoi des identificateurs de toutes les offres de délégation que le courtier connaît. Cette opération est mappée sur l'opération de recherche X.500. La donnée est la suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
how_many (nombre)	Non utilisé	Ce paramètre est conservé par l'agent T-DUA dans sa mise en œuvre de l'itérateur d'identificateur

L'agent T-DUA exécute une opération de recherche X.500 pour renvoyer les identificateurs de toutes les entrées offre de délégation. Il s'agit d'une recherche à un niveau de toutes les entrées offre de délégation. Aucune information n'est demandée. Ces identificateurs (le cas échéant) sont renvoyés à l'utilisateur sous la forme suivante:

Paramètre	Syntaxe d'attribut X.500	Observation
ids (identificateurs)	Ensemble de chaînes d'annuaire	

où chaque identificateur (représentation sous forme de chaîne d'un nom distinctif partiel) identifie de manière univoque une entrée offre de délégation.

L'agent T-DUA est responsable de la mise en œuvre de l'itérateur d'identificateur.

Un contrôle d'accès est appliqué par l'agent DSA sur la base de l'information de contrôle d'accès stockée au moment de l'exportation de l'offre de délégation et seuls les administrateurs peuvent exécuter cette opération de listage des offres de délégation. Si un client a des droits d'accès insuffisants, l'erreur de sécurité insuffisantAccessRights (droits d'accès insuffisants) est renvoyée à l'agent T-DUA, cette erreur étant mappée sur l'erreur système appropriée.

## 7.9 Opérations d'évaluation de propriété dynamique

L'agent T-DUA joue le rôle de client pour les opérations d'évaluation de propriété dynamique visant à obtenir les valeurs de propriétés dynamiques. L'opération d'évaluation de propriété dynamique est la seule opération spécifiée.

### 7.9.1 Evaluation de propriété dynamique

Pour obtenir la valeur d'une propriété dynamique au moment de l'exécution, on spécifie la signature de l'opération d'évaluation de propriété dynamique. Les données de l'opération sont les suivantes:

Paramètre	Syntaxe d'attribut X.500	Observation
name (nom)	Identificateur d'objet	Identifie le nom de la propriété
returned_type (type renvoyé)	Identificateur d'objet	Identifie le type de propriété
extra_info (informations supplémentaires)	Quelconque (facultatif)	Peut comporter des informations nécessaires pour l'obtention de la valeur de propriété

L'opération renvoie la valeur associée à la propriété dont le type a la valeur du paramètre "returned\_type". L'anomalie DPEvalFailure (échec d'évaluation de propriété dynamique) est signalée si la valeur de la propriété ne peut pas être déterminée.

La façon dont la valeur d'une propriété dynamique est effectivement obtenue n'est pas spécifiée dans la Rec. UIT-T X.950 | ISO/CEI 13235-1.

## 8 Répertoire de types

Le concept de répertoire de types concerne le courtier. Un répertoire de types minimal est défini dans le présent article. Cette définition ne couvre que les parties du répertoire de types qui sont nécessaires pour permettre le fonctionnement correct de l'annuaire X.500.

NOTE 1 – Des répertoires de types plus sophistiqués sont possibles, mais les travaux dans ce domaine deviennent rapidement de la recherche en répertoires de types et ne correspondent plus à la spécification d'un courtier au moyen de la X.500.

NOTE 2 – Les travaux relatifs à la fonction de répertoire de types ODP sont en cours au moment de la publication de la présente Spécification.

### 8.1 Schéma X.500 et répertoire de types minimal

Le répertoire de types minimal correspond au schéma X.500 habituel. Les fonctions assurées par cette composante (et les limites) sont décrites ci-dessous.

Chaque identificateur de type de service est un identificateur d'objet qui identifie un type de service. A chaque type de service sont associés un type d'interface (identifié par un identificateur d'objet) et un type de propriétés de service (identifié aussi par un identificateur d'objet). Le type d'interface donne les opérations (et leurs paramètres associés) relatives au service et le type de propriétés de service donne les propriétés qui sont associées au service.

Lorsque l'agent T-DUA reçoit un identificateur de type de service de son exportateur, il envoie l'identificateur d'objet à son répertoire de types duquel il reçoit l'identificateur de type d'interface (identificateur d'objet) et l'identificateur de type de propriétés de service (identificateur d'objet) associés. L'agent T-DUA stocke l'identificateur de type de service dans l'entrée offre de service et utilise l'identificateur de type de propriétés de service comme classe d'objets auxiliaire pour cette entrée offre de service.

La classe d'objets auxiliaire donne la définition des attributs de propriété qui peuvent être inclus dans une offre de service. Pour cela, elle contient deux listes:

- la liste des attributs obligatoires, qui contient les attributs de propriété qui doivent être présents dans l'offre de service;
- la liste des attributs facultatifs, qui contient les attributs de propriété qui peuvent être présents dans l'offre de service.

Les attributs de propriété qui ne figurent dans aucune de ces deux listes ne peuvent pas être inclus dans les propriétés de service de l'offre de service. L'utilisation appropriée de règles de contrôle d'accès sur les propriétés permet de déterminer si une propriété donnée est en lecture seule et si elle peut être modifiée ou supprimée. Le traitement des propriétés dynamiques est abordé à l'article 9.

Les attributs de propriété sont aussi identifiés par des identificateurs d'objet. Les définitions des attributs permettent de contrôler:

- si plusieurs valeurs peuvent être stockées dans l'attribut;
- la syntaxe (par exemple, chaîne, entier) des valeurs. Des syntaxes d'attribut complexes (par exemple une valeur composée d'une chaîne suivie d'un entier) peuvent aussi être définies. Pour certaines syntaxes (par exemple la syntaxe des numéros de téléphone), il existe des restrictions sur les caractères qui peuvent apparaître dans la valeur;
- la façon dont les valeurs sont comparées; c'est-à-dire dans quel cas on considère que deux valeurs sont égales – il pourrait ne pas s'agir du cas où elles sont identiques.

Une limite importante concernant les définitions des attributs (par rapport aux propriétés) est qu'il est impossible de contrôler les valeurs qui doivent être acceptées pour cet attribut.

Les définitions d'attributs sont indépendantes des définitions des classes d'objets. N'importe quel attribut peut être utilisé avec n'importe quelle définition de classe d'objets. Le sous-typage de valeurs de propriété est autorisé.

Les méthodes utilisées pour l'attribution des identificateurs d'objet aident à garantir qu'un identificateur d'objet donné n'identifie pas deux types différents (types de service, types d'interface, types de propriétés de service) ou deux propriétés différentes (attributs de propriété), même parmi plusieurs courtiers appartenant à des organisations différentes. Les conflits dus au fait que deux courtiers différents ont attribué le même identificateur d'objet à deux types différents ou à deux propriétés différentes ne doivent jamais exister.

Les définitions de types et de propriétés sont connues à la fois par l'agent DSA stockant la base de données de courtier et par l'agent T-DUA. Dans un agent DSA, le schéma X.500 est stocké sous forme d'une partie de la base de données X.500 et peut, théoriquement, être géré de façon dynamique.

## 9 Propriétés dynamiques

L'utilisation de la gestion de schéma X.500 pour mettre en œuvre une partie du répertoire de types a des incidences sur le stockage des propriétés dans l'annuaire X.500. Dans la gestion de schéma X.500, on rejette toute tentative de création d'une entrée pour laquelle il manque des attributs obligatoires (propriétés), même s'il peut s'agir de propriétés dynamiques.

La solution retenue consiste pour l'agent T-DUA à ajouter des instances d'une propriété dynamique quelconque à l'offre de service avant de la stocker dans l'annuaire X.500. Dans une instance de propriété dynamique d'une exportation, on précise que la structure `dynamicProp` est utilisée au lieu d'indiquer une valeur normale pour la propriété. Un courtier qui prend en charge une propriété dynamique reconnaît cette structure et l'accepte comme une propriété possédant une valeur dynamique. L'agent T-DUA stocke une propriété dynamique avec une valeur fictive et crée, en même temps, une valeur `dynamicPropValue` associée dans l'attribut `dynamicProps` qui contient les informations nécessaires à l'agent T-DUA pour obtenir une valeur de propriété d'un type correct lorsque c'est nécessaire. La valeur fictive de l'attribut de propriété correspondant est ignorée pendant la recherche d'offres de service concordantes. Une valeur `dynamicPropValue` distincte est définie pour chaque propriété dynamique éventuelle devant être stockée dans le courtier.

Supposons par exemple qu'un exportateur exporte une offre de service avec la propriété "queueLength" (longueur de file d'attente) indiquée comme étant dynamique. Lorsqu'il crée l'entrée offre de service dans la base de données X.500, l'agent T-DUA crée une instance de la propriété "queueLength" et la valeur `dynamicPropValue` 'queueLengthDynamicPropValue' (valeur de propriété dynamique pour la longueur de la file d'attente) dans cette entrée. Au moment de la lecture de cette entrée offre de service, la présence de la valeur 'queueLengthDynamicPropValue' dans l'attribut `dynamicProps` indique que la valeur de l'attribut correspondant (propriété) "queueLength" figurant dans la base de données X.500 doit être ignorée et que la valeur réelle doit être obtenue à partir de l'interface donnée dans la valeur `dynamicPropValue`.

Une autre solution consisterait à inclure le commutateur 'dynamique' dans la définition de chaque attribut (c'est-à-dire que tout attribut possède une valeur réelle ou le fanion 'dynamique' positionné). Cette solution a été rejetée car toutes les définitions d'attribut et l'ensemble du traitement seraient alors compliqués.

### 9.1 Exportation d'une offre de service

Au moment de l'exportation d'une offre de service, le client spécifie l'identificateur d'objet du type de service et un ensemble de propriétés (attributs). Si le courtier prend en charge des propriétés dynamiques et que de telles propriétés sont indiquées dans l'offre de service, l'agent T-DUA crée des valeurs fictives et l'attribut `dynamicProps` avec une valeur `dynamicPropValue` pour chaque propriété dynamique de l'offre de service. L'agent T-DUA positionne aussi l'attribut

hasDynamicProperties sur 'true' (vrai). Si le courtier ne prend pas en charge de propriété dynamique, toute offre de service comportant des propriétés dynamiques est rejetée. L'annuaire X.500 rejette aussi l'exportation si l'offre de service ne cadre pas avec le schéma.

Si le client souhaite exporter une offre dont le type de service est nouveau, l'exportation devient un processus à deux étapes. Toutes les nouvelles définitions de type et de propriété sont d'abord ajoutées au schéma X.500 conservé dans le répertoire de types. Une fois les nouveaux types et attributs (propriétés) enregistrés avec l'agent DSA, le client peut exporter la nouvelle offre de service.

## 9.2 Importation d'une offre de service

Au moment de l'importation d'une offre de service, le client spécifie le type de service (identificateur d'objet) qu'il souhaite trouver et une contrainte de concordance.

Une forme limitée de formation de sous-classe est disponible dans la X.500. Une classe d'objets peut être définie comme étant une sous-classe d'une ou de plusieurs classes d'objets existantes. Dans ce cas, la spécification du type de propriétés de service (identificateur d'objet) d'une superclasse permettra aussi de sélectionner les offres de service de la sous-classe. La limite associée au concept de superclasse/sous-classe concerne la liste des attributs obligatoires et la liste des attributs facultatifs de la définition de la classe d'objets. Une sous-classe hérite de la liste des attributs obligatoires et de la liste des attributs facultatifs de ses superclasses et peut uniquement contenir des attributs supplémentaires.

L'agent T-DUA recherche dans la base de données X.500 les offres de service et les offres de délégation concordantes (si c'est nécessaire) d'abord pour le type de service requis et recherche ensuite les offres concordantes pour la contrainte et le type requis.

Si le courtier prend en charge des propriétés dynamiques et que l'importateur demande l'utilisation de propriétés dynamiques, la valeur de l'attribut hasDynamicProperties est également demandée à la première recherche. Si cet attribut a la valeur 'true' (vrai), le filtre relatif à la contrainte de concordance est réécrit en conséquence. Pour chaque entrée renvoyée, l'agent T-DUA parcourt la liste des attributs renvoyés. En cas de présence d'une valeur dynamicPropValue, l'agent T-DUA demande la valeur réelle de la propriété associée à partir de l'interface donnée dans la valeur dynamicPropValue. Une fois que les propriétés dynamiques ont été renvoyées, l'agent T-DUA doit terminer l'évaluation de la contrainte de concordance.

## Annexe A

## Définition schématique des définitions de courtier

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

```

TraderDefinitions {joint-iso-itu-t 2}
DEFINITIONS ::=
BEGIN
IMPORTS
    informationFramework, selectedAttributeTypes, authenticationFramework
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
                                usefulDefinitions(0) 2}

    CONTENT-RULE, NAME-FORM, STRUCTURE-RULE, OBJECT-CLASS, MATCHING-RULE,
    ATTRIBUTE, top, ObjectClassKind, objectIdentifierMatch,
    DistinguishedName
        FROM InformationFramework informationFramework

    DirectoryString {}, commonName, description, presentationAddress,
    distinguishedName, caseIgnoreMatch, caseIgnoreSubstringsMatch,
    caseExactMatch, booleanMatch, integerMatch, octetStringMatch
        FROM SelectedAttributeTypes selectedAttributeTypes

    userPassword
        FROM AuthenticationFramework authenticationFramework ;

-- Trader Entry
traderEntry OBJECT-CLASS ::= {
    SUBCLASS OF          { top }
    MUST CONTAIN         {commonName | traderInterface | dsaName |
                          typeRepos | defSearchCard | maxSearchCard |
                          defMatchCard | maxMatchCard | defReturnCard |
                          maxReturnCard | defHopCount | maxHopCount |
                          defFollowPolicy | maxFollowPolicy |
                          maxLinkFollowPolicy | supportsModifiableProperties |
                          supportsDynamicProperties | supportsProxyOffers |
                          maxList | requestIdStem}
    MAY CONTAIN          {description | userPassword}
    ID                   id-trader-oc-traderEntry}

traderInterface ATTRIBUTE ::= {
    SUBTYPE OF           presentationAddress
    SINGLE VALUE         TRUE
    ID                   id-trader-at-traderInterface}

```

```

dsaName ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-dsaName}

typeRepos ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-typeRepos}

defSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defSearchCard }

maxSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxSearchCard}

defMatchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defMatchCard}

maxMatchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxMatchCard}

defReturnCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defReturnCard}

maxReturnCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxReturnCard}

```

## ISO/CEI 13235-3 : 1998 (F)

```
defHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defHopCount}

maxHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxHopCount}

defFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defFollowPolicy }

FollowOption ::= ENUMERATED{
    localOnly    (0),
    ifNoLocal    (1),
    always       (2)}

maxFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxFollowPolicy }

maxLinkFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxLinkFollowPolicy }

supportsModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsModifiableProperties}

supportsDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsDynamicProperties}
```

```

supportsProxyOffers ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-supportsProxyOffers}

maxList ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxList }

requestIdStem ATTRIBUTE ::= {
    WITH SYNTAX                OCTET STRING (SIZE (0..ub-request-id-stem))
    EQUALITY MATCHING RULE     octetStringMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-requestIdStem }

--Trader Policy Entry

traderPolicyEntryNF NAME-FORM ::= {
    NAMES                       traderPolicyEntry
    WITH ATTRIBUTES             {commonName}
    ID                          id-trader-nf-traderPolicy}

traderPolicyEntry OBJECT-CLASS ::= {
    SUBCLASS OF                 {top}
    MUST CONTAIN                 {commonName }
    MAY CONTAIN                  {typeManagementConstraint | searchConstraint |
                                offerAcceptanceConstraint }
    ID                           id-trader-oc-traderPolicy}
PolicySpecification ::= CHOICE {
    stringRule                   [0]   DirectoryString{ub-policy-string-rule},
    policyObjectId               [1]   DistinguishedName }

policySpecificationMatch MATCHING-RULE ::= {
    SYNTAX                       PolicySpecification
    ID                            id-trader-mr-policySpecificationMatch}

-- The rule returns TRUE if two specifications contain exactly
-- the same characters.

typeManagementConstraint ATTRIBUTE ::= {
    WITH SYNTAX                  PolicySpecification
    EQUALITY MATCHING RULE       policySpecificationMatch
    SINGLE VALUE                 TRUE
    ID                            id-trader-at-typeManagementConstraint}

```

## ISO/CEI 13235-3 : 1998 (F)

```
searchConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-searchConstraint}

offerAcceptanceConstraint ATTRIBUTE ::= {
    WITH SYNTAX                PolicySpecification
    EQUALITY MATCHING RULE     policySpecificationMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-offerAcceptanceConstraint}

-- Service Offer Entry

serviceOfferEntryNF NAME-FORM ::= {
    NAMES                      serviceOfferEntry
    WITH ATTRIBUTES            {sOfferId}
    ID                          id-trader-nf-serviceOffer}
serviceOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF                {top}
    MUST CONTAIN                {sOfferId | serviceInterfaceId | serviceTypeId
                                hasDynamicProperties | hasModifiableProperties }
    MAY CONTAIN                 {dynamicProps}
    ID                          id-trader-oc-serviceOffer}

sOfferId ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString{ub-s-offer-id}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-sOfferId}

serviceInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF                 distinguishedName
    SINGLE VALUE               TRUE
    ID                          id-trader-at-serviceInterfaceId}

serviceTypeId ATTRIBUTE ::= {
    WITH SYNTAX                OBJECT IDENTIFIER
    EQUALITY MATCHING RULE     objectIdentifierMatch
    ID                          id-trader-at-serviceTypeId }

hasDynamicProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-hasDynamicProperties}
```

```

hasModifiableProperties ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-hasModifiableProperties}

dynamicProps ATTRIBUTE ::= {
    WITH SYNTAX                SEQUENCE OF DynamicPropValue
    ID                          id-trader-at-dynamicProps}

DynamicPropValue ::= SEQUENCE {
    propertyType                OBJECT IDENTIFIER,
    dynamicPropEvalIf          DistinguishedName,
    extraInfo                   DirectoryString{ub-dynamic-value-extra-info}}

dynamicPropValueMatch MATCHING-RULE ::= {
    SYNTAX                      DynamicPropValue
    ID                          Id-trader-mr-dynamicPropValueMatch}

-- The rule returns TRUE if two values contain exactly the same characters.
-- Trader Link Entry

traderLinkEntryNF NAME-FORM ::= {
    NAMES                       traderLinkEntry
    WITH ATTRIBUTES             {linkId}
    ID                          id-trader-nf-traderLink}

traderLinkEntry OBJECT-CLASS ::= {
    SUBCLASS OF                 {top}
    MUST CONTAIN                 {linkName | linkId | targetTraderInterfaceId |
                                defPassOnFollowRule | limitingFollowRule}
    ID                          id-trader-oc-traderLink}

linkName ATTRIBUTE ::= {
    WITH SYNTAX                 DirectoryString {ub-link-name}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-linkName}

linkId ATTRIBUTE ::= {
    WITH SYNTAX                 DirectoryString {ub-link-id}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-linkId}

```

**ISO/CEI 13235-3 : 1998 (F)**

```
targetTraderInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-targetTraderInterfaceId}

defPassOnFollowRule ATTRIBUTE ::= {
    WITH SYNTAX         FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defPassOnFollowRule }

limitingFollowRule ATTRIBUTE ::= {
    WITH SYNTAX         FollowOption
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-limitingFollowRule }

-- Proxy Offer Entry

proxyOfferEntryNF NAME-FORM ::= {
    NAMES               proxyOfferEntry
    WITH ATTRIBUTES    {proxyOfferId}
    ID                  id-trader-nf-proxyOffer}

proxyOfferEntry OBJECT-CLASS ::= {
    SUBCLASS OF        {top}
    MUST CONTAIN       {proxyOfferId | proxyLookUpInterfaceId |
                       hasDynamicProperties | hasModifiableProperties |
                       ifMatchAll | constraintRecipe}
    MAY CONTAIN        { dynamicProps }
    ID                  id-trader-oc-proxyOffer}

proxyOfferId ATTRIBUTE ::= {
    WITH SYNTAX        DirectoryString {ub-proxy-offer-id}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-proxyOfferId}

proxyLookUpInterfaceId ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-proxyLookUpInterfaceId}
```

```

constraintRecipe ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-constraint-recipe}
    EQUALITY MATCHING RULE  caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE          TRUE
    ID                    id-trader-at-constraintRecipe}

ifMatchAll ATTRIBUTE ::= {
    WITH SYNTAX          BOOLEAN
    EQUALITY MATCHING RULE  booleanMatch
    SINGLE VALUE          TRUE
    ID                    id-trader-at-ifMatchAll }

-- Interface Entry

interfaceEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    KIND                  auxiliary
    MUST CONTAIN          {interfaceReference | interfaceType}
    ID                    id-trader-oc-interfaceEntry}

interfaceReference ATTRIBUTE ::= {
    WITH SYNTAX          InterfaceId
    EQUALITY MATCHING RULE  caseExactMatch
    ID                    id-trader-at-interfaceReference}

InterfaceId ::=          DirectoryString {ub-interface-id}

interfaceType ATTRIBUTE ::= {
    WITH SYNTAX          InterfaceTypeName
    EQUALITY MATCHING RULE  objectIdentifierMatch
    ID                    id-trader-at-interfaceType}

InterfaceTypeName ::=    OBJECT IDENTIFIER

-- Object Identifier Assignments

id-trader                OBJECT IDENTIFIER ::= {joint-iso-itu-t
trader(100)}

id-trader-at             OBJECT IDENTIFIER ::= {id-trader 4}
id-trader-oc             OBJECT IDENTIFIER ::= {id-trader 6}
id-trader-mr             OBJECT IDENTIFIER ::= {id-trader 13}
id-trader-nf             OBJECT IDENTIFIER ::= {id-trader 15}

id-trader-oc-traderEntry OBJECT IDENTIFIER ::= {id-trader-oc 0}
id-trader-oc-serviceOffer OBJECT IDENTIFIER ::= {id-trader-oc 1}
id-trader-oc-proxyOffer  OBJECT IDENTIFIER ::= {id-trader-oc 2}

```

**ISO/CEI 13235-3 : 1998 (F)**

id-trader-oc-traderLink	OBJECT IDENTIFIER ::= {id-trader-oc 3}
id-trader-oc-traderPolicy	OBJECT IDENTIFIER ::= {id-trader-oc 4}
id-trader-oc-interfaceEntry	OBJECT IDENTIFIER ::= {id-trader-oc 5}
id-trader-nf-serviceOffer	OBJECT IDENTIFIER ::= {id-trader-nf 1}
id-trader-nf-traderLink	OBJECT IDENTIFIER ::= {id-trader-nf 2}
id-trader-nf-traderPolicy	OBJECT IDENTIFIER ::= {id-trader-nf 3}
id-trader-nf-proxyOffer	OBJECT IDENTIFIER ::= {id-trader-nf 4}
id-trader-at-traderInterface	OBJECT IDENTIFIER ::= {id-trader-at 0}
id-trader-at-typeRepos	OBJECT IDENTIFIER ::= {id-trader-at 1}
id-trader-at-defSearchCard	OBJECT IDENTIFIER ::= {id-trader-at 2}
id-trader-at-maxSearchCard	OBJECT IDENTIFIER ::= {id-trader-at 3}
id-trader-at-defMatchCard	OBJECT IDENTIFIER ::= {id-trader-at 4}
id-trader-at-maxMatchCard	OBJECT IDENTIFIER ::= {id-trader-at 5}
id-trader-at-commonName	OBJECT IDENTIFIER ::= {id-trader-at 6}
id-trader-at-dsaName	OBJECT IDENTIFIER ::= {id-trader-at 7}
id-trader-at-defReturnCard	OBJECT IDENTIFIER ::= {id-trader-at 10}
id-trader-at-maxReturnCard	OBJECT IDENTIFIER ::= {id-trader-at 11}
id-trader-at-defHopCount	OBJECT IDENTIFIER ::= {id-trader-at 12}
id-trader-at-maxHopCount	OBJECT IDENTIFIER ::= {id-trader-at 13}
id-trader-at-defFollowPolicy	OBJECT IDENTIFIER ::= {id-trader-at 14}
id-trader-at-maxLinkFollowPolicy	OBJECT IDENTIFIER ::= {id-trader-at 15}
id-trader-at-maxFollowPolicy	OBJECT IDENTIFIER ::= {id-trader-at 16}
id-trader-at-supportsModifiableProperties	OBJECT IDENTIFIER ::= {id-trader-at 20}
id-trader-at-supportsDynamicProperties	OBJECT IDENTIFIER ::= {id-trader-at 21}
id-trader-at-supportsProxyOffers	OBJECT IDENTIFIER ::= {id-trader-at 22}
id-trader-at-maxList	OBJECT IDENTIFIER ::= {id-trader-at 23}
id-trader-at-requestIdStem	OBJECT IDENTIFIER ::= {id-trader-at 24}
id-trader-at-typeManagementConstraint	OBJECT IDENTIFIER ::= {id-trader-at 25}
id-trader-at-searchConstraint	OBJECT IDENTIFIER ::= {id-trader-at 30}
id-trader-at-offerAcceptanceConstraint	OBJECT IDENTIFIER ::= {id-trader-at 31}
id-trader-at-sOfferId	OBJECT IDENTIFIER ::= {id-trader-at 32}
id-trader-at-serviceTypeId	OBJECT IDENTIFIER ::= {id-trader-at 33}
id-trader-at-serviceInterfaceId	OBJECT IDENTIFIER ::= {id-trader-at 34}
id-trader-at-hasDynamicProperties	OBJECT IDENTIFIER ::= {id-trader-at 35}
id-trader-at-hasModifiableProperties	OBJECT IDENTIFIER ::= {id-trader-at 40}
id-trader-at-dynamicProps	OBJECT IDENTIFIER ::= {id-trader-at 41}
id-trader-at-linkId	OBJECT IDENTIFIER ::= {id-trader-at 42}
id-trader-at-linkName	OBJECT IDENTIFIER ::= {id-trader-at 43}
id-trader-at-targetTraderInterfaceId	OBJECT IDENTIFIER ::= {id-trader-at 44}
id-trader-at-defPassOnFollowRule	OBJECT IDENTIFIER ::= {id-trader-at 45}
id-trader-at-limitingFollowRule	OBJECT IDENTIFIER ::= {id-trader-at 50}
id-trader-at-proxyOfferId	OBJECT IDENTIFIER ::= {id-trader-at 51}
id-trader-at-proxyLookUpInterfaceId	OBJECT IDENTIFIER ::= {id-trader-at 52}
id-trader-at-constraintRecipe	OBJECT IDENTIFIER ::= {id-trader-at 53}

```

id-trader-at-ifMatchAll          OBJECT IDENTIFIER ::= {id-trader-at 55}
id-trader-at-interfaceReference  OBJECT IDENTIFIER ::= {id-trader-at 60}
id-trader-at-interfaceType      OBJECT IDENTIFIER ::= {id-trader-at 61}

```

```

id-trader-mr-policySpecificationMatch  OBJECT IDENTIFIER ::= {id-trader-mr 1}
id-trader-mr-dynamicPropValueMatch    OBJECT IDENTIFIER ::= {id-trader-mr 2}

```

-- *Upperbounds*

```

ub-common-name                INTEGER ::= 64
ub-request-id-stem           INTEGER ::= 1024
ub-policy-string-rule        INTEGER ::= 1024
ub-s-offer-id                INTEGER ::= 64
ub-dynamic-value-extra-info  INTEGER ::= 1024
ub-link-name                  INTEGER ::= 64
ub-link-id                    INTEGER ::= 64
ub-proxy-offer-id            INTEGER ::= 64
ub-constraint-recipe         INTEGER ::= 1024
ub-interface-id              INTEGER ::= 1024

```

END

## Annexe B

## Exemple de définition schématique de description de service

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

```

PrinterServiceOfferDefinitions {joint-iso-itu-t 2}
DEFINITIONS ::=
BEGIN
IMPORTS
    informationFramework, selectedAttributeTypes
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) modules(1)
            usefulDefinitions(0) 2}

OBJECT-CLASS, MATCHING-RULE, ATTRIBUTE, top, ObjectClassKind, DistinguishedName
    FROM InformationFramework informationFramework

DirectoryString {}, caseIgnoreMatch, caseIgnoreSubstringsMatch, caseExactMatch,
booleanMatch, integerMatch, integerOrderMatch
    FROM SelectedAttributeTypes selectedAttributeTypes

id-trader-at, id-trader-oc-serviceOffer
    FROM id-trader{joint-iso-itu-t trader(100)} ;

printerServiceOffer OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    KIND                 auxiliary
    MUST CONTAIN        printerType}
    MAY CONTAIN          {locationRoom | locationBuilding | costPerPage |
        languagesSupported | pagesPerMinute | pageSize |
        dotsPerInch | colourCapable | driverName | queueLength}
    ID                  id-trader-oc-serviceOffer-printer}

printerType ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-trader-so-printerType}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-so-printerType}

locationRoom ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-trader-so-locationRoom}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-so-locationRoom }

```

```
locationBuilding ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-trader-so-locationBlg}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-locationBlg}
```

```
costPerPage ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-costPerPage}
```

```
languagesSupported ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-trader-so-langSupp}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-langSupp}
```

```
pagesPerMinute ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-pagesPerMinute}
```

```
pagesPerMinute ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-pagesPerMinute}
```

```
dotsPerInch ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE              TRUE
    ID                        id-trader-at-so-dotsPerInch}
```

## ISO/CEI 13235-3 : 1998 (F)

```
colourCapable ATTRIBUTE ::= {
    WITH SYNTAX                BOOLEAN
    EQUALITY MATCHING RULE     booleanMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-so-colourCapable}

driverName ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-trader-so-driverName}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE  caseIgnoreSubstringsMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-so-driverName}

queueLength ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    ORDERING MATCHING RULE    integerOrderMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-so-queueLength }

-- Object Identifiers

id-trader-oc-serviceOffer-printer
                                OBJECT IDENTIFIER ::= {id-trader-oc-serviceOffer 0}

id-trader-at-so                 OBJECT IDENTIFIER ::= {id-trader-at 100}
id-trader-at-so-printerType     OBJECT IDENTIFIER ::= {id-trader-at-so 0}
id-trader-at-so-locationRoom    OBJECT IDENTIFIER ::= {id-trader-at-so 1}
id-trader-at-so-locationBlg    OBJECT IDENTIFIER ::= {id-trader-at-so 2}
id-trader-at-so-costPerPage     OBJECT IDENTIFIER ::= {id-trader-at-so 3}
id-trader-at-so-langSupp       OBJECT IDENTIFIER ::= {id-trader-at-so 4}
id-trader-at-so-pagesPerMinute OBJECT IDENTIFIER ::= {id-trader-at-so 5}
id-trader-at-so-pageSize       OBJECT IDENTIFIER ::= {id-trader-at-so 6}
id-trader-at-so-dotsPerInch    OBJECT IDENTIFIER ::= {id-trader-at-so 7}
id-trader-at-so-colourCapable  OBJECT IDENTIFIER ::= {id-trader-at-so 8}
id-trader-at-so-driverName     OBJECT IDENTIFIER ::= {id-trader-at-so 9}
id-trader-at-so-queueLength    OBJECT IDENTIFIER ::= {id-trader-at-so 10}

-- Upperbounds

ub-trader-so-printerType       INTEGER ::= 64
ub-trader-so-locationRoom     INTEGER ::= 64
ub-trader-so-locationBlg      INTEGER ::= 64
ub-trader-so-langSupp         INTEGER ::= 64
ub-trader-so-pageSize         INTEGER ::= 64
ub-trader-so-driverName       INTEGER ::= 64

END
```

## SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
<b>Série X</b>	<b>Réseaux pour données et communication entre systèmes ouverts</b>
Série Y	Infrastructure mondiale de l'information
Série Z	Langages de programmation