



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.520

(02/2001)

SÉRIE X: RÉSEAUX DE DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Annuaire

**Technologies de l'information – Interconnexion
des systèmes ouverts – L'annuaire: types
d'attributs sélectionnés**

Recommandation UIT-T X.520

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE X
RÉSEAUX DE DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS DE DONNÉES	
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
INTERCONNEXION DES SYSTÈMES OUVERTS	
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 des couches	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
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
GESTION OSI	
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
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

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

Technologies de l'information – Interconnexion des systèmes ouverts –
L'annuaire: types d'attributs sélectionnés

Résumé

La présente Recommandation | Norme internationale définit un certain nombre de types d'attributs et de règles de concordance pouvant s'avérer utiles dans différentes applications de l'annuaire. Nombre de ces attributs ainsi définis peuvent notamment servir à la formation de noms, en particulier pour les classes d'objets définies dans la Rec. UIT-T X.521 | ISO/CEI 9594-7.

Source

La Recommandation X.520 de l'UIT-T, élaborée par la Commission d'études 7 (2001-2004) de l'UIT-T, a été approuvée le 2 février 2001. Un texte identique est publié comme Norme Internationale ISO/CEI 9594-6.

Note

Les réalisateurs et les utilisateurs sont priés de noter qu'il existe un processus de résolution des erreurs et que des corrections peuvent être apportées à la présente Recommandation | Norme internationale sous la forme de corrigenda techniques. Des corrections identiques peuvent aussi être apportées à la présente Recommandation sous la forme d'un Guide d'implémentation. Une liste des corrigenda techniques qui ont été approuvés pour la présente Norme internationale figure sur le site web de l'ISO. Pour obtenir les corrigenda techniques qui ont été publiés, veuillez vous adresser à l'organisme de normalisation de votre pays. Les corrigenda techniques et le Guide d'implémentation pour la présente Recommandation figurent sur le site web de l'UIT-T.

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.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude 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 1 de l'AMNT.

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 2002

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'applicatio	1
2	Références normatives	1
	2.1 Recommandations Normes internationales identiques	1
	2.2 Autres références	2
	2.3 Normes ISO/CEI	2
3	Définitions	2
4	Conventions	3
5	Définition des types d'attributs sélectionnés	4
	5.1 Attributs de type système	4
	5.2 Attributs de type étiquetage	4
	5.3 Attributs de type géographique	6
	5.4 Attributs de type organisation	8
	5.5 Attributs de type explicatif	9
	5.6 Attributs de type adresse postale	11
	5.7 Attributs de type adresse pour télécommunication	12
	5.8 Attributs de type priorité	16
	5.9 Attributs de type applications OSI	16
	5.10 Attributs de type relationnel	17
	5.11 Attributs de type domaine	18
	5.12 Attributs de notification	18
6	Définition des règles de concordance	24
	6.1 Règles de concordance par chaînes	24
	6.2 Règles de concordance à base syntaxique	27
	6.3 Time matching rules (règles de concordance à base temporelle)	30
	6.4 First component matching rules (règles de concordance de premier élément)	31
	6.5 Word matching rules (règles de concordance de mots)	31
	6.6 Règles de correspondance approchée (correspondance approchée des chaînes)	34
	6.7 Règles de correspondance spéciales	34
	6.8 Zonal Match (correspondance zonale)	35
7	Définition des types de contextes	40
	7.1 Language Context (contexte linguistique)	40
	7.2 Temporal Context (contexte temporel)	40
	7.3 Locale Context (contexte de localité)	43
	Annexe A – Types d'attributs sélectionnés en ASN.1	45
	Annexe B – Résumé des types d'attributs	63
	Annexe C – Limites supérieures	65
	Annexe D – Index alphabétique des attributs, des règles de concordance et des contextes	66
	Annexe E – Exemples de règles de correspondance zonale	68
	Annexe F – Amendements et corrigenda	71

Introduction

La présente Recommandation | Norme internationale a été élaborée, ainsi que d'autres Recommandations | Normes internationales, pour faciliter l'interconnexion des systèmes de traitement de l'information et permettre ainsi d'assurer des services d'annuaire. L'ensemble de tous ces systèmes, avec les informations d'annuaire qu'ils contiennent, peut être considéré comme un tout intégré, appelé *annuaire*. Les informations de l'annuaire, appelées collectivement base d'informations d'annuaire (DIB), sont généralement utilisées pour faciliter la communication entre, avec ou à propos d'objets tels que des entités d'application, des personnes, des terminaux et des listes de distribution.

L'annuaire joue un rôle important dans l'interconnexion des systèmes ouverts, dont le but est de permettre, moyennant un minimum d'accords techniques en dehors des normes d'interconnexion proprement dites, l'interconnexion des systèmes de traitement de l'information:

- provenant de divers fabricants;
- gérés différemment;
- de niveaux de complexité différents;
- de générations différentes.

La présente Recommandation | Norme internationale définit un certain nombre de types d'attribut qui peuvent être jugés utiles dans une gamme d'applications de l'annuaire ainsi qu'un certain nombre de syntaxes d'attribut et de règles de concordance normales. Un grand nombre des attributs ici définis sert en particulier à former des noms, notamment pour les classes d'objets définies dans la Rec. UIT-T X.521 | ISO/CEI 9594-7.

Cette quatrième édition révisé techniquement et améliore, mais ne remplace pas, la troisième édition de la présente Recommandation | Norme internationale. Les implémentations peuvent encore revendiquer la conformité à la troisième édition mais celle-ci finira par ne plus être prise en compte (c'est-à-dire que les erreurs signalées ne seront plus corrigées). Il est recommandé que les implémentations se conforment, dès que possible, à la présente quatrième édition.

Cette quatrième édition spécifie les versions 1 et 2 des protocoles de l'annuaire.

Les première et deuxième éditions ne spécifiaient que la version 1. La plupart des services et protocoles spécifiés dans la présente édition sont conçus pour fonctionner selon la version 1. Certains services et protocoles améliorés, par exemple les erreurs signées, ne fonctionneront cependant pas avant que toutes les entités d'annuaire mises en jeu dans l'exploitation aient négocié la version 2. Quelle que soit la version négociée, on a traité les différences entre les services et entre les protocoles, définis dans les quatre éditions, à l'exception de ceux qui sont spécifiquement définis dans la version 2, en utilisant les règles d'extensibilité définies dans l'édition actuelle de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

L'Annexe A, qui fait partie intégrante de la présente Recommandation | Norme internationale, fournit la notation ASN.1 du module complet qui définit les attributs, la syntaxe des attributs et les règles de concordance.

L'Annexe B, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, propose un index des types d'attributs qui facilitera les recherches.

L'Annexe C, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, indique les valeurs des limites supérieures proposées, utilisées dans les présentes Spécifications d'annuaire.

L'Annexe D, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, donne une liste alphabétique des attributs et des règles de concordance définis dans la présente Spécification d'annuaire.

L'Annexe E, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, donne des exemples adaptés à la définition de la correspondance zonale.

L'Annexe F, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, donne une liste des amendements et rapports de défauts qui ont été incorporés pour former l'édition de la présente Recommandation | Norme internationale.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés

SECTION 1 – GÉNÉRALITÉS

1 Domaine d'application

La présente Recommandation | Norme internationale définit plusieurs types d'attributs et règles de concordance qui peuvent être utiles dans une gamme d'applications de l'annuaire.

Les types d'attributs et les règles de concordance formeront trois catégories, comme indiqué ci-dessous.

Certains types d'attributs et règles de concordance sont utilisés dans une grande variété d'applications ou sont interprétés ou utilisés par l'annuaire proprement dit.

NOTE – Il est recommandé d'utiliser les types d'attributs et règles de concordance définis dans la présente Recommandation | Norme internationale, plutôt que d'en créer de nouveaux, chaque fois que cela convient pour l'application étudiée.

Certains types d'attributs et règles de concordance sont normalisés au niveau international mais sont propres à une application. Ils sont définis dans les normes relatives à l'application en cause.

Toute autorité administrative peut définir ses propres types d'attributs et règles de concordance à une fin ou à une autre. Ces conventions ne sont pas normalisées au niveau international et ne pourront être mises que par accord bilatéral à la disposition d'autorités administratives autres que celles qui les aura créées.

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.200 (1994) | ISO/CEI 7498-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: le modèle de référence de base.*
- Recommandation UIT-T X.500 (2001) | ISO/CEI 9594-1:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: aperçu général des concepts, modèles et services.*
- Recommandation UIT-T X.501 (2001) | ISO/CEI 9594-2:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: les modèles.*
- Recommandation UIT-T X.509 (2000) | ISO/CEI 9594-8:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: cadre général des certificats de clé publique et d'attribut.*
- Recommandation UIT-T X.511 (2001) | ISO/CEI 9594-3:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: définition du service abstrait.*
- Recommandation UIT-T X.518 (2001) | ISO/CEI 9594-4:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: procédures pour le fonctionnement réparti.*

- Recommandation UIT-T X.519 (2001) | ISO/CEI 9594-5:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: spécification des protocoles.*
- Recommandation UIT-T X.521 (2001) | ISO/CEI 9594-7:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: classes d'objets sélectionnées.*
- Recommandation UIT-T X.525 (2001) | ISO/CEI 9594-9:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: duplication.*
- Recommandation UIT-T X.530 (2001) | ISO/CEI 9594-10:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: utilisation de la gestion-systèmes pour l'administration de l'annuaire.*
- Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- Recommandation UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels.*
- Recommandation UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes.*
- Recommandation UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un.*

2.2 Autres références

- Recommandation UIT-T E.123 (2001), *Notation des numéros téléphoniques nationaux et internationaux, des adresses de courrier électronique et des adresses web.*
- Recommandation UIT-T E.164 (1997), *Plan de numérotage des télécommunications publiques internationales.*
- Recommandation UIT-T F.1 (1998), *Dispositions applicables à l'exploitation du service public international des télégrammes.*
- Recommandation CCITT F.31 (1988), *Système à retransmission de télégrammes.*
- Recommandation CCITT F.401 (1992), *Dénomination et adressage pour les services publics de messagerie.*
- Recommandation UIT-T T.30 (1996), *Procédures pour la transmission de documents par télécopie sur le réseau téléphonique public commuté.*
- Recommandation UIT-T T.62 (1993), *Procédures de commande pour le service télétexte et le service de télécopie du Groupe 4.*
- Recommandation UIT-T X.121 (2000), *Plan de numérotage international pour les réseaux publics de données.*
- ISO 3166 (toutes les parties), *Codes pour la représentation des noms de pays et de leurs subdivisions.*
- ISO 639-2:1998, *Codes pour la représentation des noms de langue – Partie 2: Code alpha-3.*
- ISO/CEI 9945-2:1993, *Technologies de l'information – Interface pour la portabilité des systèmes (POSIX) – Partie 2: Enveloppe et services.*

2.3 Normes ISO/CEI

- ISO/CEI 10646-1:2000, *Technologies de l'information – Jeu universel de caractères codés sur plusieurs octets (JUC) – Partie 1: Architecture et plan multilingue de base.*

3 Définitions

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

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

- a) *type d'attribut;*
- b) *classe d'objets;*
- c) *règle de concordance;*
- d) *contexte.*

4 Conventions

A quelques exceptions mineures près, la présente Spécification d'annuaire a été élaborée conformément aux "Règles de présentation des textes communs UIT-T | ISO/CEI", qui figurent dans le Guide relatif à la coopération entre l'UIT-T et l'ISO/CEI JTC 1, octobre 1996.

Le terme "Spécification d'annuaire" (comme dans "la présente Spécification d'annuaire") s'entend selon l'acception de la présente Recommandation | Norme internationale. Le terme "Spécification d'annuaire" s'entend selon l'acception de toutes les Recommandations de la série X.500 | toutes les parties de l'ISO/CEI 9594.

La présente Spécification d'annuaire utilise le terme "systèmes de l'édition 1988" pour désigner les systèmes conformes à la première édition (1988) des Spécifications d'annuaire, c'est-à-dire à l'édition 1988 des Recommandations CCITT de la série X.500 et de l'ISO/CEI 9594:1990. La présente Spécification d'annuaire utilise le terme "systèmes de l'édition 1993" pour désigner les systèmes conformes à la deuxième édition (1993) des Spécifications d'annuaire, c'est-à-dire l'édition 1993 des Recommandations UIT-T de la série X.500 et de l'ISO/CEI 9594:1995. La présente Spécification d'annuaire utilise le terme "systèmes de l'édition 1997" pour désigner les systèmes conformes à la troisième édition des Spécifications d'annuaire, c'est-à-dire à l'édition 1997 des Recommandations UIT-T de la série X.500 et de l'ISO/CEI 9594:1998. La présente Spécification d'annuaire utilise le terme "systèmes de la quatrième édition" pour désigner les systèmes conformes à la présente quatrième édition des Spécifications d'annuaire, c'est-à-dire aux éditions 2001 des Recommandations UIT-T X.500, X.501, X.511, X.518, X.519, X.520, X.521, X.525 et X.530, à l'édition 2000 de la Recommandation UIT-T X.509, et aux parties 1 à 10 de l'ISO/CEI 9594:2001.

Cette Spécification d'annuaire présente la notation ASN.1 en caractères gras de la police Helvetica. Lorsque des types et des valeurs ASN.1 sont cités dans le texte normal, ils en sont différenciés par leur présentation en caractères gras Helvetica. Les noms des procédures, normalement cités lors de la spécification des sémantèmes de traitement, sont différenciés du texte normal par une présentation en caractères gras de la police Times. Les autorisations de contrôle d'accès sont présentées en caractères italiques de la police Times.

Les types d'attributs, les règles de concordance et les types de contexte sont définis dans la présente Recommandation | Norme internationale par les classes d'objets d'information **ATTRIBUTE** (attribut), **MATCHING-RULE** (règle de concordance) et **CONTEXT** définies dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.

Les exemples d'utilisation des types d'attributs font intervenir une notation non formelle dans laquelle les paires de types et de valeurs d'attributs sont représentées par un acronyme désignant le type d'attribut, suivi d'un signe d'égalité (=) et de la valeur de l'attribut donnée comme exemple.

SECTION 2 – TYPES D'ATTRIBUTS SÉLECTIONNÉS

5 Définition des types d'attributs sélectionnés

La présente Spécification d'annuaire définit un certain nombre de types d'attributs pouvant être utiles dans une large gamme d'applications de l'annuaire.

Un grand nombre des attributs définis dans la présente Spécification sont fondés sur une syntaxe ASN.1 commune:

```
DirectoryString { INTEGER : maxSize } ::= CHOICE {
    teletexString      TeletexString (SIZE (1..maxSize)),
    printableString    PrintableString (SIZE (1..maxSize)),
    bmpString          BMPString (SIZE (1..maxSize)),
    universalString    UniversalString (SIZE (1..maxSize)),
    uTF8String         UTF8String (SIZE (1..maxSize)) }
```

Certaines implémentations de l'annuaire peuvent ne pas gérer les chaînes de type **UniversalString**, **BMPString** ou **UTF8String**. Elles peuvent donc ne pas être en mesure de produire, de mettre en correspondance, de dupliquer ou d'afficher les attributs ayant ces types de syntaxe.

5.1 Attributs de type système

5.1.1 Knowledge Information (informations de connaissance)

Le type d'attribut *Knowledge Information* spécifie une description accumulée, accessible en lecture par l'homme, d'une connaissance maîtrisée par un DSA donné.

NOTE – Cet attribut n'est plus utilisé.

```
knowledgeInformation ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-knowledge-information}
    EQUALITY MATCHING RULE caseIgnoreMatch
    ID                   id-at-knowledgeInformation }
```

5.2 Attributs de type étiquetage

Ces types d'attributs concernent l'information relative à des objets qui a été explicitement associée à ces objets par un processus d'étiquetage.

5.2.1 Name (nom)

Le type d'attribut *Name* est le supertype d'attribut à partir duquel on peut former les attributs de type chaîne normalement utilisés pour la dénomination.

```
name ATTRIBUTE ::= {
    WITH SYNTAX          DirectoryString {ub-name}
    EQUALITY MATCHING RULE caseIgnoreMatch
    SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
    ID                   id-at-name }
```

5.2.2 Common Name (nom commun)

Le type d'attribut *Common Name* spécifie l'identificateur d'un objet. Le Common Name n'est pas un nom d'annuaire, mais le nom (parfois ambigu) sous lequel l'objet est généralement connu dans un cercle assez restreint (par exemple une organisation). Il est conforme aux conventions de dénomination du pays ou de la culture auxquels il est associé.

Pour Common Name, une valeur d'attributs est une chaîne choisie par la personne ou par l'organisation qu'elle décrit, ou par l'organisation responsable de l'objet décrit pour des dispositifs et des entités d'application. Par exemple, le nom typique d'une personne se trouvant dans un pays anglophone comprend un titre personnel (par exemple Mr., Ms., Rd, Professor, Sir, Lord), un prénom, un ou plusieurs autres prénoms, un nom de famille, une indication de génération (par exemple, le cas échéant, Jr.), les titres et des décorations (par exemple, le cas échéant, QC).

Exemples

CN = "Mr. Robin Lachlan McLeod BSc(Hons) CEng MIEE";

CN = "Divisional Coordination Committee";

CN = "High Speed Modem".

Toutes les variantes doivent être associées à l'objet nommé sous la forme de valeurs d'attributs séparées et secondaires.

D'autres variantes communes doivent aussi être admises, par exemple l'emploi du second prénom au lieu du premier, le remplacement de "William" par "Bill", etc.

```
commonName ATTRIBUTE ::= {
    SUBTYPE OF          name
    WITH SYNTAX        DirectoryString {ub-common-name}
    ID                  id-at-commonName }
```

5.2.3 Surname (nom de famille)

Le type d'attribut *Surname* spécifie la construction linguistique dont une personne hérite en général de ses parents ou qu'elle prend lors de son mariage et sous laquelle elle est généralement connue.

Pour Surname, une valeur d'attribut est une chaîne, par exemple "McLeod".

```
surname ATTRIBUTE ::= {
    SUBTYPE OF          name
    WITH SYNTAX        DirectoryString {ub-surname}
    ID                  id-at-surname }
```

5.2.4 Given Name (prénom)

Le type d'attribut *Given Name* spécifie la forme linguistique normalement donnée à une personne par ses parents, ou choisie par elle, ou encore par laquelle ladite personne est généralement connue.

Pour Given Name, une valeur d'attribut est une chaîne, par exemple "David" ou "Jean-Paul".

```
givenName ATTRIBUTE ::= {
    SUBTYPE OF          name
    WITH SYNTAX        DirectoryString {ub-name}
    ID                  id-at-givenName }
```

5.2.5 Initials (initiales)

Le type d'attribut *Initials* contient les initiales de certains des noms d'une personne ou de tous ses noms à l'exclusion du nom de famille.

Pour Initials, une valeur d'attribut est une chaîne, par exemple "D" ou "D." ou "J.P.".

```
initials ATTRIBUTE ::= {
    SUBTYPE OF          name
    WITH SYNTAX        DirectoryString {ub-name}
    ID                  id-at-initials }
```

5.2.6 Generation Qualifier (qualificateur de génération)

Le type d'attribut *Generation Qualifier* contient une chaîne utilisée pour fournir des informations de génération qualifiant le nom d'une personne.

Pour Generation Qualifier, une valeur d'attribut est une chaîne, par exemple "Jr." ou "II".

```
generationQualifier ATTRIBUTE ::= {
    SUBTYPE OF          name
    WITH SYNTAX        DirectoryString {ub-name}
    ID                  id-at-generationQualifier }
```

5.2.7 Unique Identifier (identificateur unique)

Le type d'attribut *Unique Identifier* spécifie un identificateur qui peut être utilisé pour faire la distinction entre des références d'objet lorsqu'un nom distinctif a été réutilisé. Il peut s'agir par exemple d'un identificateur d'objet, d'un certificat, d'une date, d'un timbre ou de toute autre forme de certification, codé(e), de la validité du nom distinctif.

Pour Unique Identifier, une valeur d'attribut est une chaîne de bits.

```
uniqueIdentifier ATTRIBUTE ::= {
    WITH SYNTAX                UniqueIdentifier
    EQUALITY MATCHING RULE     bitStringMatch
    ID                          id-at-uniqueIdentifier }
```

UniqueIdentifier ::= BIT STRING

5.2.8 DN Qualifier (qualificateur DN)

Le type d'attribut *DN Qualifier* spécifie des informations de levée d'ambiguïté à ajouter au nom distinctif relatif d'une entrée. Il doit être utilisé pour des entrées relevant de DSA multiples qui autrement auraient le même nom et, dans un DSA donné, doit avoir la même valeur pour toutes les entrées auxquelles ces informations ont été ajoutées.

```
dnQualifier ATTRIBUTE ::= {
    WITH SYNTAX                PrintableString
    EQUALITY MATCHING RULE     caseIgnoreMatch
    ORDERING MATCHING RULE     caseIgnoreOrderingMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    ID                          id-at-dnQualifier }
```

5.2.9 Serial Number (numéro de série)

Le type d'attribut *Serial Number* spécifie un identificateur, à savoir le numéro de série d'un objet.

Pour Serial Number, une valeur d'attribut est une chaîne imprimable.

```
serialNumber ATTRIBUTE ::= {
    WITH SYNTAX                PrintableString (SIZE (1..ub-serial-number))
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    ID                          id-at-serialNumber }
```

5.2.10 Pseudonyme

Le type d'attribut *Pseudonym* spécifie un pseudonyme pour un objet. Il sert à désigner un objet par un pseudonyme.

```
pseudonym ATTRIBUTE ::= {
    SUBTYPE OF                 name
    WITH SYNTAX                DirectoryString {ub-pseudonym}
    ID                          id-at-pseudonym }
```

5.3 Attributs de type géographique

Ces types d'attributs concernent les positions géographiques ou les régions auxquelles des objets sont associés.

5.3.1 Country Name (nom de pays)

Le type d'attribut *Country Name* spécifie un pays. Utilisé comme composant d'un nom d'annuaire, il identifie le pays dans lequel l'objet nommé est situé physiquement ou auquel cet objet est associé d'une autre manière pertinente.

Pour Country Name, une valeur d'attribut est une chaîne choisie dans ISO 3166.

```
countryName ATTRIBUTE ::= {
  SUBTYPE OF      name
  WITH SYNTAX     CountryName
  SINGLE VALUE    TRUE
  ID              id-at-countryName }
```

CountryName ::= PrintableString (SIZE(2)) -- Codes ISO 3166 seulement

5.3.2 Locality Name (nom de localité)

Le type d'attribut *Locality Name* spécifie une localité. Utilisé comme composant d'un nom d'annuaire, il identifie une zone géographique ou une localité dans laquelle l'objet nommé est situé physiquement ou à laquelle cet objet est associé d'une autre manière pertinente.

Pour Locality Name, une valeur d'attribut est une chaîne, par exemple L = "Edinburgh".

```
localityName ATTRIBUTE ::= {
  SUBTYPE OF      name
  WITH SYNTAX     DirectoryString {ub-locality-name}
  ID              id-at-localityName }
```

Le type d'attribut *Collective Locality Name* spécifie un nom de localité pour un ensemble d'entrées.

```
collectiveLocalityName ATTRIBUTE ::= {
  SUBTYPE OF      localityName
  COLLECTIVE      TRUE
  ID              id-at-collectiveLocalityName }
```

5.3.3 State or Province Name (nom d'état ou de province)

Le type d'attribut *State or Province Name* spécifie un état ou une province. Utilisé comme composant d'un nom d'annuaire, il identifie une subdivision géographique dans laquelle l'objet nommé est situé physiquement ou à laquelle cet objet est associé d'une autre manière pertinente.

Pour State or Province Name, une valeur d'attribut est une chaîne, par exemple, S = "Ohio".

```
stateOrProvinceName ATTRIBUTE ::= {
  SUBTYPE OF      name
  WITH SYNTAX     DirectoryString {ub-state-name}
  ID              id-at-stateOrProvinceName }
```

Le type d'attribut *Collective State or Province Name* spécifie un nom d'état ou de province pour un ensemble d'entrées.

```
collectiveStateOrProvinceName ATTRIBUTE ::= {
  SUBTYPE OF      stateOrProvinceName
  COLLECTIVE      TRUE
  ID              id-at-collectiveStateOrProvinceName }
```

5.3.4 Street Address (adresse de rue)

Le type d'attribut *Street Address* spécifie un emplacement pour la distribution locale et la remise physique à une adresse postale, c'est-à-dire le nom de la rue, de la place, ou de l'avenue et le numéro de la maison. Utilisé comme composant d'un nom d'annuaire, il identifie l'adresse de la rue à laquelle l'objet nommé est situé ou à laquelle cet objet est associé d'une autre manière pertinente.

Pour Street Address, une valeur d'attribut est une chaîne, par exemple " Arnulfstraße 60".

```
streetAddress ATTRIBUTE ::= {
  WITH SYNTAX     DirectoryString {ub-street-address}
  EQUALITY MATCHING RULE caselgnoreMatch
  SUBSTRINGS MATCHING RULE caselgnoreSubstringsMatch
  ID              id-at-streetAddress }
```

Le type d'attribut *Collective Street Address* spécifie une adresse de rue pour un ensemble d'entrées.

```
collectiveStreetAddress ATTRIBUTE ::= {
    SUBTYPE OF      streetAddress
    COLLECTIVE      TRUE
    ID              id-at-collectiveStreetAddress }
```

5.3.5 House Identifier (identificateur de domicile)

Le type d'attribut *House Identifier* spécifie une forme linguistique utilisée pour identifier un bâtiment, par exemple un numéro ou un nom de maison par référence à une rue, une avenue, une agglomération ou une ville, etc.

Pour House Identifier, une valeur d'attribut est une chaîne, par exemple "14".

```
houseIdentifier ATTRIBUTE ::= {
    WITH SYNTAX      DirectoryString {ub-name}
    EQUALITY MATCHING RULE caselgnoreMatch
    SUBSTRINGS MATCHING RULE caselgnoreSubstringsMatch
    ID              id-at-houseIdentifier }
```

5.4 Attributs de type organisation

Ces types d'attributs concernent les organisations et peuvent servir à décrire des objets d'après les organisations auxquelles ils sont associés.

5.4.1 OrganizationName (nom d'organisation)

Le type d'attribut *OrganizationName* spécifie une organisation. Utilisé comme composant d'un nom d'annuaire, il identifie une organisation à laquelle l'objet nommé est affilié.

Pour **OrganizationName**, une valeur d'attribut est une chaîne choisie par l'organisation (par exemple 0 = "Scottish Telecommunications plc"). Toute variante doit être associée à l'organisation nommée sous la forme de valeurs d'attribut séparées et secondaires.

```
organizationName ATTRIBUTE ::= {
    SUBTYPE OF      name
    WITH SYNTAX      DirectoryString {ub-organization-name}
    ID              id-at-organizationName }
```

Le type d'attribut *Collective Organization Name* spécifie un nom d'organisation pour un ensemble d'entrées.

```
collectiveOrganizationName ATTRIBUTE ::= {
    SUBTYPE OF      organizationName
    COLLECTIVE      TRUE
    ID              id-at-collectiveOrganizationName }
```

5.4.2 Organizational Unit Name (nom d'unité d'organisation)

Le type d'attribut *Organizational Unit Name* spécifie une unité d'organisation. Utilisé comme composant d'un nom d'annuaire, il identifie une unité d'organisation à laquelle l'objet nommé est affilié.

On considère que l'unité d'organisation désignée fait partie d'une organisation désignée par un attribut **organizationName**. En conséquence, si un attribut *Organizational Unit Name* est utilisé dans un nom d'annuaire, il doit être associé à un attribut **organizationName**.

Pour Organizational Unit Name, une valeur d'attribut est une chaîne choisie par l'organisation dont l'unité fait partie (par exemple OU = "Technology Division"). On notera que l'abréviation "TD", communément utilisée, serait une valeur d'attribut séparée et secondaire.

Exemple

O = "Scottel", OU = "TD"

```
organizationalUnitName ATTRIBUTE ::= {
  SUBTYPE OF name
  WITH SYNTAX DirectoryString {ub-organizational-unit-name}
  ID id-at-organizationalUnitName }
```

Le type d'attribut *Collective Organizational Unit Name* spécifie un nom d'unité d'organisation pour un ensemble d'entrées.

```
collectiveOrganizationalUnitName ATTRIBUTE ::= {
  SUBTYPE OF organizationalUnitName
  COLLECTIVE TRUE
  ID id-at-collectiveOrganizationalUnitName }
```

5.4.3 Title (titre)

Le type d'attribut *Title* spécifie la position ou la fonction désignée de l'objet dans une organisation.

Pour Title, une valeur d'attribut est une chaîne.

Exemple

T = "Manager, Distributed Applications"

```
title ATTRIBUTE ::= {
  SUBTYPE OF name
  WITH SYNTAX DirectoryString {ub-title}
  ID id-at-title }
```

5.5 Attributs de type explicatif

Ces types d'attributs expliquent (par exemple en langage naturel) quelque chose qui concerne un objet.

5.5.1 Description

Le type d'attribut *Description* spécifie un texte qui décrit l'objet associé.

Par exemple, à l'objet "participation aux normes" peut être associée la description "liste de distribution pour l'échange d'information concernant l'élaboration des normes dans la compagnie".

Pour Description, une valeur d'attribut est une chaîne.

```
description ATTRIBUTE ::= {
  WITH SYNTAX DirectoryString {ub-description}
  EQUALITY MATCHING RULE caseIgnoreMatch
  SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
  ID id-at-description }
```

5.5.2 Search Guide (guide de recherche)

Le type d'attribut *Search Guide* spécifie l'information des critères de recherche suggérés. Ces critères peuvent être inclus dans certaines entrées, considérées comme objets de base utiles pour l'opération de recherche, par exemple un pays ou une organisation.

Les critères de recherche comprennent un identificateur facultatif pour le type d'objet recherché et des combinaisons de types d'attributs et d'opérateurs logiques à utiliser pour la construction d'un filtre. Il est possible de spécifier, pour chaque élément de critère de recherche, le niveau de concordance, par exemple une concordance approximative.

L'attribut Search Guide peut se répéter pour tenir compte des divers types de demande, par exemple une recherche de Residential Person ou d'Organizational Person, qui peuvent être accomplis à partir de l'objet de base donné où Search Guide est lu.

```
searchGuide ATTRIBUTE ::= {
  WITH SYNTAX Guide
  ID id-at-searchGuide }
```

```
Guide ::= SET {
  objectClass      [0]  OBJECT-CLASS.&id OPTIONAL,
  criteria          [1]  Criteria }
```

```
Criteria ::= CHOICE {
  type      [0]  Criterialtem,
  and       [1]  SET OF Criteria,
  or        [2]  SET OF Criteria,
  not       [3]  Criteria }
```

```
Criterialtem ::= CHOICE {
  equality      [0]  AttributeType,
  substrings   [1]  AttributeType,
  greaterOrEqual [2]  AttributeType,
  lessOrEqual  [3]  AttributeType,
  approximateMatch [4]  AttributeType }
```

Exemple

On trouvera ci-après une valeur possible de l'attribut Search Guide qui peut être stocké dans des entrées de la classe d'objets Locality pour indiquer comment il est possible de trouver des entrées de la classe d'objets Residential Person.

```
residential-person-guide Guide ::= {
  objectClass residentialPerson.&id,
  criteria and : {
    type : substrings : commonName.&id,
    type : substrings : streetAddress.&id } }
```

La construction d'un filtre à partir de cette valeur de guide est simple.

L'étape (1) produit la valeur Filter intermédiaire:

```
intermediate-filter Filter ::=
  and : {
    item : substrings {
      type commonName.&id,
      strings { any : teletexString : "Dubois" } },
    item : substrings {
      type streetAddress.&id,
      strings { any : teletexString "Hugo" } } }
```

L'étape (2) produit un filtre pour mettre en correspondance les entrées Residential Person dans le sous-arbre:

```
residential-person-filter Filter ::=
  and : {
    item : equality : {
      type objectClass.&id,
      assertion residentialPerson.&id },
    intermediateFilter }
```

5.5.3 Enhanced Search Guide (guide de recherche amélioré)

Le type d'attribut *Enhanced Search Guide* constitue une amélioration de l'attribut **searchGuide** du fait qu'il ajoute des renseignements sur la profondeur de recherche recommandée pour des recherches entre objets subordonnés d'une classe d'objets donnée.

```
enhancedSearchGuide ATTRIBUTE ::= {
  WITH SYNTAX EnhancedGuide
  ID          id-at-enhancedSearchGuide }
```

```
EnhancedGuide ::= SEQUENCE {
  objectClass [0]  OBJECT-CLASS.&id,
  criteria    [1]  Criteria,
  subset      [2]  INTEGER
  { baseObject (0), oneLevel (1), wholeSubtree (2) } DEFAULT oneLevel }
```


5.5.4 Business Category (catégorie d'occupation)

L'attribut *Business Category* spécifie l'information relative à l'occupation de certains objets communs, tels que des personnes. Par exemple, cet attribut permet d'interroger l'annuaire au sujet de personnes partageant la même occupation.

```
businessCategory ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-business-category}
    EQUALITY MATCHING RULE     caselgnoreMatch
    SUBSTRINGS MATCHING RULE   caselgnoreSubstringsMatch
    ID                          id-at-businessCategory }
```

5.6 Attributs de type adresse postale

Ces types d'attributs concernent l'information nécessaire pour la remise postale physique à un objet.

5.6.1 Postal Address (adresse postale)

Le type d'attribut *Postal Address* spécifie l'information d'adresse nécessaire pour la remise physique des messages postaux par l'autorité postale à l'objet nommé.

Pour *Postal Address*, une valeur d'attribut sera généralement composée d'attributs choisis à partir de l'adresse postale O/R non formatée – version 1 du système de messagerie, conformément à la Rec. CCITT F.401 et limitée à 6 lignes de 30 caractères comprenant un nom de code postal. Normalement, l'information contenue dans une telle adresse peut comprendre un nom de destinataire, nom et numéro de la rue, ville, état ou province, code postal et éventuellement un numéro de boîte postale, en fonction des besoins propres à l'objet nommé.

```
postalAddress ATTRIBUTE ::= {
    WITH SYNTAX                PostalAddress
    EQUALITY MATCHING RULE     caselgnoreListMatch
    SUBSTRINGS MATCHING RULE   caselgnoreListSubstringsMatch
    ID                          id-at-postalAddress }
```

```
PostalAddress ::= SEQUENCE SIZE(1..ub-postal-line) OF DirectoryString {ub-postal-string}
```

Le type d'attribut *Collective Postal Address* spécifie une adresse postale pour un ensemble d'entrées.

```
collectivePostalAddress ATTRIBUTE ::= {
    SUBTYPE OF                postalAddress
    COLLECTIVE                 TRUE
    ID                          id-at-collectivePostalAddress }
```

5.6.2 Postal Code (code postal)

Le type d'attribut *Postal Code* spécifie le code postal de l'objet nommé. Si cette valeur d'attribut est présente, elle fait partie de l'adresse postale de l'objet.

Pour *Postal Code*, une valeur d'attribut est une chaîne.

```
postalCode ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-postal-code}
    EQUALITY MATCHING RULE     caselgnoreMatch
    SUBSTRINGS MATCHING RULE   caselgnoreSubstringsMatch
    ID                          id-at-postalCode }
```

Le type d'attribut *Collective Postal Code* spécifie un code postal pour un ensemble d'entrées.

```
collectivePostalCode ATTRIBUTE ::= {
    SUBTYPE OF                postalCode
    COLLECTIVE                 TRUE
    ID                          id-at-collectivePostalCode }
```

5.6.3 Post Office Box (boîte postale)

Le type d'attribut *Post Office Box* spécifie la boîte postale par laquelle l'objet recevra la remise postale physique. Si elle est présente, la valeur d'attribut fait partie de l'adresse postale de l'objet.

```
postOfficeBox ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-post-office-box}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    ID                          id-at-postOfficeBox }
```

Le type d'attribut *Collective Post Office Box* spécifie une boîte postale pour un ensemble d'entrées.

```
collectivePostOfficeBox ATTRIBUTE ::= {
    SUBTYPE OF      postOfficeBox
    COLLECTIVE      TRUE
    ID              id-at-collectivePostOfficeBox }
```

5.6.4 Physical Delivery Office Name (nom du bureau de remise physique)

Le type d'attribut *Physical Delivery Office Name* spécifie le nom de la ville, du village, etc., où se trouve un bureau de remise physique.

Pour *Physical Delivery Office Name*, une valeur d'attribut est une chaîne.

```
physicalDeliveryOfficeName ATTRIBUTE ::= {
    WITH SYNTAX                DirectoryString {ub-physical-office-name}
    EQUALITY MATCHING RULE     caseIgnoreMatch
    SUBSTRINGS MATCHING RULE   caseIgnoreSubstringsMatch
    ID                          id-at-physicalDeliveryOfficeName }
```

Le type d'attribut *Collective Physical Delivery Office Name* spécifie un nom de bureau de remise physique pour un ensemble d'entrées.

```
collectivePhysicalDeliveryOfficeName ATTRIBUTE ::= {
    SUBTYPE OF      physicalDeliveryOfficeName
    COLLECTIVE      TRUE
    ID              id-at-collectivePhysicalDeliveryOfficeName }
```

5.7 Attributs de type adresse pour télécommunication

Ces types d'attributs concernent l'information d'adressage nécessaire pour communiquer avec l'objet en utilisant des moyens de télécommunication.

5.7.1 Telephone Number (numéro de téléphone)

Le type d'attribut *Telephone Number* spécifie un numéro de téléphone associé à un objet.

Pour *Telephone Number*, une valeur d'attribut est une chaîne qui correspond au format international normalisé pour représenter les numéros de téléphone internationaux (Rec. UIT-T E.123), par exemple "+ 44 582 10101".

```
telephoneNumber ATTRIBUTE ::= {
    WITH SYNTAX                TelephoneNumber
    EQUALITY MATCHING RULE     telephoneNumberMatch
    SUBSTRINGS MATCHING RULE   telephoneNumberSubstringsMatch
    ID                          id-at-telephoneNumber }
```

```
TelephoneNumber ::= PrintableString (SIZE(1..ub-telephone-number))
    -- Chaîne en conformité avec la Rec. UIT-T E.123 seulement
```

L'attribut du type *Collective Telephone Number* spécifie un numéro de téléphone pour un ensemble d'entrées.

```
collectiveTelephoneNumber ATTRIBUTE ::= {
  SUBTYPE OF   telephoneNumber
  COLLECTIVE   TRUE
  ID           id-at-collectiveTelephoneNumber }
```

5.7.2 Telex Number (numéro de télex)

Le type d'attribut *Telex Number* spécifie le numéro télex, l'indicatif de pays et l'indicatif d'un terminal télex associé à un objet.

```
telexNumber ATTRIBUTE ::= {
  WITH SYNTAX TelexNumber
  ID          id-at-telexNumber }
```

```
TelexNumber ::= SEQUENCE {
  telexNumber      PrintableString (SIZE (1..ub-telex-number)),
  countryCode      PrintableString (SIZE (1..ub-country-code)),
  answerback      PrintableString (SIZE (1..ub-answerback)) }
```

Le type d'attribut *Collective Telex Number* spécifie un numéro télex pour un ensemble d'entrées.

```
collectiveTelexNumber ATTRIBUTE ::= {
  SUBTYPE OF   telexNumber
  COLLECTIVE   TRUE
  ID           id-at-collectiveTelexNumber }
```

5.7.3 Teletex Terminal Identifier (identificateur de terminal télétex)

*Etant donné que la Rec. CCITT F.200 a été supprimée sans être remplacée, l'utilisation des types d'attributs **teletexTerminalIdentifier** et **collectiveTeletexTerminalIdentifier** est déconseillée.*

Le type d'attribut *Teletex Terminal Identifier* spécifie l'identificateur (et, facultativement, les paramètres) du terminal télétex associé à un objet.

Pour Teletex Terminal Identifier, une valeur d'attribut est une chaîne qui satisfait aux spécifications de la Rec. CCITT F.200 et un ensemble facultatif dont les composants sont conformes à la Rec. UIT-T T.62.

```
-- teletexTerminalIdentifier ATTRIBUTE ::= {
--   WITH SYNTAX TeletexTerminalIdentifier
--   ID          id-at-teletexTerminalIdentifier }

-- TeletexTerminalIdentifier ::= SEQUENCE {
--   teletexTerminal      PrintableString (SIZE(1..ub-teletex-terminal-id)),
--   parameters          TeletexNonBasicParameters OPTIONAL }
```

Le type d'attribut *Collective Teletex Terminal Identifier* spécifie un identificateur de terminal télétex pour un ensemble d'entrées.

```
-- collectiveTeletexTerminalIdentifier ATTRIBUTE ::= {
--   SUBTYPE OF   teletexTerminalIdentifier
--   COLLECTIVE   TRUE
--   ID          id-at-collectiveTeletexTerminalIdentifier }
```

5.7.4 Facsimile Telephone Number (numéro téléphonique de télécopie)

Le type d'attribut *Facsimile Telephone Number* spécifie un numéro de téléphone pour un terminal de télécopie (et, facultativement, ses paramètres) associé à un objet.

ISO/CEI 9594-6:2001 (F)

Pour Facsimile Telephone Number, une valeur d'attribut est une chaîne qui satisfait au format international convenu pour représenter les numéros de téléphone internationaux (Rec. UIT-T E.123), par exemple "+81 3 347 7418" et une chaîne de bits facultative (formatée selon la Rec. UIT-T T.30).

```
facsimileTelephoneNumber ATTRIBUTE ::= {  
    WITH SYNTAX    FacsimileTelephoneNumber  
    ID              id-at-facsimileTelephoneNumber }
```

```
FacsimileTelephoneNumber ::= SEQUENCE {  
    telephoneNumberTelephoneNumber,  
    parameters          G3FacsimileNonBasicParameters OPTIONAL }
```

Le type d'attribut *Collective Facsimile Telephone Number* spécifie un numéro téléphonique de télécopie pour un ensemble d'entrées.

```
collectiveFacsimileTelephoneNumber ATTRIBUTE ::= {  
    SUBTYPE OF     facsimileTelephoneNumber  
    COLLECTIVE     TRUE  
    ID              id-at-collectiveFacsimileTelephoneNumber }
```

5.7.5 X.121 Address (adresse X.121)

Le type d'attribut *X.121 Address* spécifie une adresse, telle que définie dans la Rec. UIT-T X.121 et associée à un objet.

```
x121Address ATTRIBUTE ::= {  
    WITH SYNTAX          X121Address  
    EQUALITY MATCHING RULE    numericStringMatch  
    SUBSTRINGS MATCHING RULE    numericStringSubstringsMatch  
    ID                      id-at-x121Address }
```

```
X121Address ::= NumericString (SIZE(1..ub-x121-address))  
-- Chaîne selon définition dans la Rec. UIT-T X.121
```

5.7.6 International ISDN Number (numéro RNIS international)

Le type d'attribut *International ISDN Number* spécifie un numéro RNIS international associé à un objet.

Pour International ISDN Number, une valeur d'attribut est une chaîne qui satisfait au format international convenu pour les adresses RNIS et présenté dans la Rec. UIT-T E.164.

```
internationalISDNNumber ATTRIBUTE ::= {  
    WITH SYNTAX          InternationalISDNNumber  
    EQUALITY MATCHING RULE    numericStringMatch  
    SUBSTRINGS MATCHING RULE    numericStringSubstringsMatch  
    ID                      id-at-internationalISDNNumber }
```

```
InternationalISDNNumber ::= NumericString (SIZE(1..ub-international-isdn-number))  
-- Chaîne en conformité avec la Rec. UIT-T E.164 seulement
```

Le type d'attribut *Collective International ISDN Number* spécifie un numéro RNIS international pour un ensemble d'entrées.

```
collectiveInternationalISDNNumber ATTRIBUTE ::= {  
    SUBTYPE OF     internationalISDNNumber  
    COLLECTIVE     TRUE  
    ID              id-at-collectiveInternationalISDNNumber }
```

5.7.7 Registered Address (adresse enregistrée)

Le type d'attribut *Registered Address* spécifie un mnémonique pour une adresse associée à un objet à l'emplacement d'une ville donnée. Le mnémonique est enregistré dans le pays où se trouve la ville et il est utilisé pour la fourniture du service public des télégrammes (selon la Rec. UIT-T F.1).

```
registeredAddress ATTRIBUTE ::= {
  SUBTYPE OF      postalAddress
  WITH SYNTAX     PostalAddress
  ID              id-at-registeredAddress }
```

5.7.8 Destination Indicator (indicateur de destination)

Le type d'attribut *Destination Indicator* spécifie (selon la Rec. UIT-T F.1 et la Rec. CCITT F.31) le pays et la ville associés à l'objet (le destinataire) et nécessaires pour fournir le service public des télégrammes.

Pour Destination Indicator, une valeur d'attribut est une chaîne.

```
destinationIndicator ATTRIBUTE ::= {
  WITH SYNTAX           DestinationIndicator
  EQUALITY MATCHING RULE caseIgnoreMatch
  SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
  ID                    id-at-destinationIndicator }
```

```
DestinationIndicator ::= PrintableString (SIZE(1..ub-destination-indicator))
-- caractères alphabétiques seulement
```

5.7.9 Service de communication

Le type d'attribut *Communication Service* spécifie le type de service(s) associé à une adresse de communication.

```
communicationsService ATTRIBUTE ::= {
  WITH SYNTAX           CommunicationsService
  EQUALITY MATCHING RULE objectIdentifierMatch
  ID                    id-at-communicationsService }
```

CommunicationsService ::= OBJECT IDENTIFIER

Cet attribut décrit la classe de service à laquelle l'adresse de communication donne accès. Par exemple, téléphone (voix), télécopie, courrier électronique, SMS (service de messages courts), EDI, transfert de fichiers, etc.

L'attribution d'identificateurs d'objet pour désigner des services s'effectuera hors de la présente spécification d'annuaire.

5.7.10 Réseau de communication

Le type d'attribut *Communication Network* spécifie le type de réseau pour lequel une adresse de communication est utilisée.

```
communicationsNetwork ATTRIBUTE ::= {
  WITH SYNTAX           CommunicationsNetwork
  EQUALITY MATCHING RULE objectIdentifierMatch
  SINGLE VALUE         TRUE
  ID                    id-at-communicationsNetwork }
```

CommunicationsNetwork ::= OBJECT IDENTIFIER

Cet attribut décrit le type de réseau dans lequel l'adresse de communication est attribuée. Par exemple, un réseau téléphonique public commuté (RTPC), un RNIS ou un réseau GSM de téléphonie mobile. Il peut s'agir également d'un réseau orienté vers une application, comme un réseau de banque à domicile.

L'attribution d'identificateurs d'objet pour désigner des réseaux s'effectuera hors de la présente spécification d'annuaire.

5.8 Attributs de type priorité

Ces types d'attributs concernent les priorités d'un objet.

5.8.1 Preferred Delivery Method (méthode de remise préférée)

Le type d'attribut *Preferred Delivery Method* spécifie l'ordre de priorité de l'objet au sujet de la méthode à utiliser pour communiquer avec lui.

```
preferredDeliveryMethod ATTRIBUTE ::= {
    WITH SYNTAX          PreferredDeliveryMethod
    SINGLE VALUE        TRUE
    ID                   id-at-preferredDeliveryMethod }
```

```
PreferredDeliveryMethod ::= SEQUENCE OF INTEGER {
    any-delivery-method (0),
    mhs-delivery        (1),
    physical-delivery   (2),
    telex-delivery      (3),
    teletex-delivery    (4),
    g3-facsimile-delivery (5),
    g4-facsimile-delivery (6),
    ia5-terminal-delivery (7),
    videotex-delivery   (8),
    telephone-delivery  (9) }
```

5.9 Attributs de type applications OSI

Ces types d'attributs concernent l'information relative aux objets dans la couche Application de l'OSI.

5.9.1 Presentation Address (adresse de présentation)

Le type d'attribut *Presentation Address* spécifie une adresse de présentation associée à un objet représentant une entité d'application OSI.

Pour Presentation Address, une valeur d'attribut est une adresse de présentation telle que définie dans la Rec. UIT-T X.200 | ISO/CEI 7498-1.

```
presentationAddress ATTRIBUTE ::= {
    WITH SYNTAX          PresentationAddress
    EQUALITY MATCHING RULE presentationAddressMatch
    SINGLE VALUE        TRUE
    ID                   id-at-presentationAddress }
```

```
PresentationAddress ::= SEQUENCE {
    pSelector [0] OCTET STRING OPTIONAL,
    sSelector [1] OCTET STRING OPTIONAL,
    tSelector [2] OCTET STRING OPTIONAL,
    nAddresses [3] SET SIZE (1..MAX) OF OCTET STRING }
```

5.9.2 Supported Application Context (contexte d'application géré)

Le type d'attribut *Supported Application Context* spécifie l'identificateur/les identificateurs d'objet d'un ou de contextes d'application que l'objet (une entité d'application OSI) peut gérer.

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

5.9.3 Protocol Information (informations sur le protocole)

Le type d'attribut *Protocol Information* associe des informations sur le protocole à chaque adresse réseau présente dans l'attribut Presentation Address.

Pour chaque **nAddress**, le composant de protocole identifie le protocole ou le profil pour les couches Réseau et Transport.

```

protocolInformation ATTRIBUTE ::= {
    WITH SYNTAX                ProtocolInformation
    EQUALITY MATCHING RULE    protocolInformationMatch
    ID                          id-at-protocolInformation }

```

```

ProtocolInformation ::= SEQUENCE {
    nAddress          OCTET STRING,
    profiles         SET OF OBJECT IDENTIFIER }

```

5.10 Attributs de type relationnel

Ces types d'attributs concernent une information relative aux objets qui, d'une certaine façon, sont associés à un objet particulier.

NOTE – La syntaxe **DistinguishedName** (nom distinctif) utilisée dans les attributs de cette catégorie permet d'employer soit un nom distinctif primaire, soit une variante de nom distinctif. L'usage, s'il est connu, du nom distinctif primaire assure la cohérence et l'interfonctionnement avec les DSA antérieurs à 1997. Il peut être demandé, dans certains cas spécifiques, que soit employée une variante particulière de nom distinctif. Il est aussi possible de conserver l'information contextuelle et les variantes de valeurs distinctives au sein du composant **valuesWithContext** de n'importe quel RDN, comme l'explique le § 9.3 de la Rec. UIT-T X.501 | ISO/CEI 9594-2.

5.10.1 Distinguished Name (nom distinctif)

Le type d'attribut *Distinguished Name* spécifie le nom d'un objet.

```

distinguishedName ATTRIBUTE ::= {
    WITH SYNTAX                DistinguishedName
    EQUALITY MATCHING RULE    distinguishedNameMatch
    ID                          id-at-distinguishedName }

```

5.10.2 Member (membre)

Le type d'attribut *Member* spécifie un groupe de noms associés à l'objet.

Pour Member, une valeur d'attribut est un nom spécifique.

```

member ATTRIBUTE ::= {
    SUBTYPE OF                distinguishedName
    ID                          id-at-member }

```

5.10.3 Unique Member (membre unique)

Le type d'attribut *Unique Member* spécifie un groupe de noms uniques associés à un objet. Un nom unique est un nom ne pouvant faire l'objet d'aucune ambiguïté lorsqu'un identificateur unique lui est, facultativement, associé.

Pour Unique Member, une valeur d'attribut est un nom distinctif accompagné d'un identificateur unique facultatif.

```

uniqueMember ATTRIBUTE ::= {
    WITH SYNTAX                NameAndOptionalUID
    EQUALITY MATCHING RULE    uniqueMemberMatch
    ID                          id-at-uniqueMember }

```

```

NameAndOptionalUID ::= SEQUENCE {
    dn          DistinguishedName,
    uid        Uniquentifier OPTIONAL }

```

5.10.4 Owner (propriétaire)

Le type d'attribut *Owner* spécifie le nom d'un objet qui a quelque responsabilité à l'égard de l'objet associé.

Pour Owner, une valeur d'attribut est un nom distinctif (qui peut représenter un groupe de noms); elle peut se répéter.

```
owner ATTRIBUTE ::= {
    SUBTYPE OF distinguishedName
    ID id-at-owner }
```

5.10.5 Role Occupant (occupant d'un rôle)

Le type d'attribut *Role Occupant* spécifie le nom d'un objet qui assume un rôle dans l'organisation.

Pour Role Occupant, une valeur d'attribut est un nom distinctif.

```
RoleOccupant ATTRIBUTE ::= {
    SUBTYPE OF distinguishedName
    ID id-at-roleOccupant }
```

5.10.6 See Also (voir également)

Le type d'attribut *See Also* spécifie les noms d'autres objets d'annuaire qui peuvent être d'autres aspects (dans un certain sens) du même objet concret.

Pour See Also, une valeur d'attribut est un nom distinctif.

```
seeAlso ATTRIBUTE ::= {
    SUBTYPE OF distinguishedName
    ID id-at-seeAlso }
```

5.11 Attributs de type domaine

5.11.1 DMD name (nom de DMD)

Le type d'attribut *DMD Name* spécifie un domaine de gestion d'annuaire. Lorsqu'il est employé comme composant d'un nom d'annuaire, il identifie un domaine de gestion d'annuaire qui gère l'objet dénommé.

Pour DMD Name, une valeur d'attribut est une chaîne choisie par le DMD.

```
dmdName ATTRIBUTE ::= {
    SUBTYPE OF name
    WITH SYNTAX DirectoryString{ub-common-name}
    ID id-at-dmdName }
```

5.12 Attributs de notification

Les attributs des notifications ont la syntaxe des attributs mais sont définis de façon à acheminer des informations additionnelles dans des éléments **CommonResults** (ou **CommonResultsSeq**) et **PartialOutcomeQualifier** (comme décrit aux § 7.4 et 10.1 de la Rec. UIT-T X.511 | ISO/CEI 9594-3). Ils sont habituellement définis avec des règles de correspondance permettant de vérifier les valeurs renvoyées en fonctions de valeurs connues localement.

5.12.1 Problème des agents DSA

L'attribut de message *DSA Problem* est utilisé conjointement avec une erreur **serviceError** ou l'élément **PartialOutcomeQualifier**. Il est défini comme suit:

```
dSAProblem ATTRIBUTE ::= {
    WITH SYNTAX OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID id-not-dSAProblem }
```


Les valeurs définies pour l'élément **dsaProblem** sont les suivantes:

- a) **id-pr-targetDsaUnavailable** – Une demande doit être concaténée à un autre agent DSA au cours de la résolution de nom, mais aucune association ne peut être établie avec cet agent DSA.
- b) **id-pr-dataSourceUnavailable** – Un agent DSA ne peut pas effectuer d'opération car cette partie de la base DIB n'est pas disponible.
- c) **id-pr-administratorImposedLimit** – Une opération a dépassé certaines limites fixées par l'administrateur.
- d) **id-pr-permanentRestriction** – Une opération a conduit un agent DSA à dépasser certaines limites, provoquant l'arrêt du processus, et il est jugé que la répétition de l'opération rencontrera le même problème.
- e) **id-pr-temporaryRestriction** – Une opération a conduit un agent DSA à dépasser certaines limites, provoquant l'arrêt du processus, et il est jugé que le motif est un problème temporaire, par exemple l'épuisement des ressources.

5.12.2 Problème du service de recherche

L'attribut de notification *Search Service Problem* décrit des problèmes lors de l'application de politiques de règle de recherche. Il est utilisé en même temps que les erreurs de service ou l'élément **PartialOutcomeQualifier**. Il est défini comme suit:

```
searchServiceProblem ATTRIBUTE ::= {
    WITH SYNTAX                OBJECT IDENTIFIER
    EQUALITY MATCHING RULE     objectIdentifierMatch
    SINGLE VALUE               TRUE
    ID                         id-not-searchServiceProblem }
```

Les valeurs définies pour l'attribut **searchServiceProblem** sont les suivantes:

- a) **id-pr-unidentifiedOperation** – L'opération tentée ne correspond pas à une de celles qui ont été identifiées pour ce service.
- b) **id-pr-unavailableOperation** – L'opération tentée n'est conforme qu'à une règle de recherche qui n'est pas disponible au demandeur.
- c) **id-pr-searchAttributeViolation** – Un ou plusieurs types d'attribut dont la présence est requise dans le filtre n'étaient pas présents.
- d) **id-pr-searchAttributeCombinationViolation** – Le filtre de la demande de recherche comprenait bien la combinaison requise de types d'attribut.
- e) **id-pr-searchValueNotAllowed** – Des valeurs d'attribut étaient spécifiées pour des types d'attribut dans lesquels seuls les types d'attribut peuvent être spécifiés dans les types d'élément de filtrage **present** et **contextPresent**.
- f) **id-pr-missingSearchAttribute** – Les attributs identifiés, qui n'étaient pas présents dans la recherche demandée, sont requis pour la règle de recherche applicable.
- g) **id-pr-searchValueViolation** – Les valeurs d'attribut identifiées pour les types d'attribut identifiés ne sont pas autorisées lors d'une recherche utilisant la règle de recherche applicable.
- h) **id-pr-attributeNegationViolation** – Le type d'attribut identifié n'est pas autorisé dans la forme inversée du filtre de recherche.
- i) **id-pr-searchValueRequired** – Le type d'attribut identifié n'est pas autorisé dans un élément de filtrage n'exigeant pas de correspondance entre valeurs.
- j) **id-pr-invalidSearchValue** – Les valeurs d'attribut identifiées ne sont pas valides pour les types d'attribut identifiés pour la règle de recherche applicable.
- k) **id-pr-searchContextViolation** – Les types de contexte identifiés dans la recherche engagée ne sont pas autorisés pour le type d'attribut.
- l) **id-pr-searchContextCombinationViolation** – Les combinaisons identifiées de types de contexte, qui n'étaient pas présentes dans la recherche demandée, sont requises pour la règle de recherche applicable.
- m) **id-pr-missingSearchContext** – Les types de contexte identifiés, qui n'étaient pas présents dans la recherche demandée, sont requis pour le type d'attribut.

- n) **id-pr-searchContextValueViolation** – Les valeurs de contexte identifiées pour les types de contexte identifiés ne sont pas autorisées pour le type d'attribut.
- o) **id-pr-searchContextValueRequired** – Le type d'attribut identifié n'est pas autorisé dans un élément de filtrage n'exigeant pas de correspondance entre valeurs.
- p) **id-pr-invalidContextSearchValue** – Les valeurs d'attribut identifiées ne sont pas valides pour les types d'attribut identifiés pour la règle de recherche applicable.
- q) **id-pr-unsupportedMatchingRule** – La règle de correspondance demandée et identifiée n'est pas prise en charge.
- r) **id-pr-attributeMatchingViolation** – La règle de correspondance demandée et identifiée, ou son usage particulier, n'est pas autorisée pour les attributs identifiés pour la règle de recherche applicable.
- s) **id-pr-unsupportedMatchingUse** – La façon dont une règle de correspondance est utilisée dans un filtre de recherche n'est pas prise en charge.
- t) **id-pr-matchingUseViolation** – La façon dont une règle de correspondance suggérée est utilisée dans un filtre de recherche n'est pas autorisée, par exemple comme spécifié dans une règle de recherche.
- u) **id-pr-hierarchySelectForbidden** – La sélection hiérarchique, sauf pour **self**, n'est pas autorisée pour ce type de demande.
- v) **id-pr-invalidHierarchySelect** – Une ou plusieurs options invalides de sélection hiérarchique étaient spécifiées dans la demande.
- w) **id-pr-unavailableHierarchySelect** – Une ou plusieurs sélections hiérarchiques ne sont pas prises en charge par la réalisation.
- x) **id-pr-invalidSearchControlOptions** – Une ou plusieurs options de recherche invalides étaient spécifiées dans la demande.
- y) **id-pr-invalidServiceControlOptions** – Une ou plusieurs options de commande de service invalides étaient spécifiées dans la demande.
- z) **id-pr-searchSubsetViolation** – Le sous-ensemble de recherche demandé n'est pas autorisé pour la règle de recherche applicable.
- aa) **id-pr-unmatchedKeyAttributes** – Une règle de correspondance par mappage a été choisie, mais les éléments de filtrage pouvant ainsi être mappés ne concordaient en aucun point avec la table de mappage correspondante.
- bb) **id-pr-ambiguousKeyAttributes** – Une règle de correspondance par mappage a été choisie, mais les éléments de filtrage pouvant ainsi être mappés concordaient en plusieurs points avec la table de mappage correspondante.
- cc) **id-pr-unavailableRelaxationLevel** – L'agent DSA ne prend pas en charge un niveau d'extension d'élargissement demandé.
- dd) **id-pr-emptyHierarchySelection** – Une sélection hiérarchique était spécifiée n'ayant pas conduit au renvoi d'une entrée, bien qu'une ou plusieurs entrées concordaient avec le filtre de recherche.
- ee) **id-pr-relaxationNotSupported** – Une relaxation était spécifiée dans la demande d'utilisateur mais n'était pas prise en charge.

5.12.3 Type de service

L'attribut de notification *Service-type* indique le type de service lors d'une recherche infructueuse.

```

serviceType ATTRIBUTE ::= {
    WITH SYNTAX                OBJECT IDENTIFIER
    EQUALITY MATCHING RULE    objectIdentifierMatch
    SINGLE VALUE              TRUE
    ID                        id-not-serviceType }

```

5.12.4 Liste des types d'attribut

L'attribut de notification *Attribute Type List* donne une liste de types d'attribut permettant de préciser un problème dans un service de recherche.

```

attributeTypeList ATTRIBUTE ::= {
    WITH SYNTAX                OBJECT IDENTIFIER
    EQUALITY MATCHING RULE    objectIdentifierMatch
    ID                        id-not-attributeTypeList }

```

5.12.5 Liste des règles de correspondance

L'attribut de notification *Matching Rule List* donne une liste de règles de correspondance permettant de préciser un problème dans un service de recherche.

```
matchingRuleList ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                   id-not-matchingRuleList }
```

5.12.6 Élément de filtrage

L'attribut de notification *Filter Item* donne une liste d'éléments de filtrage invalides dans un filtre de recherche.

```
filterItem ATTRIBUTE ::= {
    WITH SYNTAX    FilterItem
    ID             id-not-filterItem }
```

5.12.7 Combinaisons d'attributs

L'attribut de notification *Attribute Combinations* donne une liste de combinaisons d'attributs dont la présentation était nécessaire dans un filtre mais qui n'ont pas été fournies.

```
attributeCombinations ATTRIBUTE ::= {
    WITH SYNTAX    AttributeCombination
    ID             id-not-attributeCombinations }
```

5.12.8 Liste des types de contexte

L'attribut de notification *Context Type List* donne une liste de types de contexte afin de préciser un problème de service de recherche.

```
contextTypeList ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                   id-not-contextTypeList }
```

5.12.9 Liste des contextes

L'attribut de notification *Context List* donne une liste de contextes afin de préciser un problème de service de recherche.

```
contextList ATTRIBUTE ::= {
    WITH SYNTAX          ContextAssertion
    ID                   id-not-contextList }
```

Une valeur de ce type d'attribut représente un type de contexte et certaines valeurs de contexte de ce type non autorisées dans la situation considérée, dont le résultat a été la production de cet attribut.

5.12.10 Combinaisons de contextes

L'attribut de notification *Context Combinations* donne une liste de combinaisons de contextes dont la présentation dans un filtre est requise.

```
contextCombinations ATTRIBUTE ::= {
    WITH SYNTAX    ContextCombination
    ID             id-not-contextCombinations }
```

5.12.11 Liste de sélection hiérarchique

L'attribut de notification *Hierarchy Select List* donne une chaîne de bits désignant une ou plusieurs options de sélection hiérarchique comme défini par la construction **HierarchySelections** qui est définie au § 10.2.1 de la Rec. UIT-T X.511 | ISO/CEI 9594-3.

```

hierarchySelectList ATTRIBUTE ::= {
    WITH SYNTAX    HierarchySelections
    SINGLE VALUE  TRUE
    ID            id-not-hierarchySelectList }

```

Lorsqu'un bit est activé dans la chaîne binaire **HierarchySelection**, il indique que la sélection hiérarchique correspondante est invalide, soit parce qu'on a demandé une sélection interdite ou non prise en charge, soit parce que la sélection n'a pas été demandée lorsqu'elle était requise.

5.12.12 Listes des options de commande de recherche

L'attribut *Search Options List* donne une chaîne de bits désignant une ou plusieurs options de commande de recherche comme défini par le type de données ASN.1 **SearchControlOptions** au § 10.2.1 de la Rec. UIT-T X.511 | ISO/CEI 9594-3.

```

searchControlOptionsList ATTRIBUTE ::= {
    WITH SYNTAX    SearchControlOptions
    SINGLE VALUE  TRUE
    ID            id-not-searchControlOptionsList }

```

Lorsqu'un bit est activé dans le type **SearchControlOptions**, il indique que la sélection d'option de commande de recherche correspondante est invalide, soit parce qu'on a demandé une option interdite ou non prise en charge, soit parce que l'option n'a pas été demandée lorsqu'elle était requise.

5.12.13 Liste des options de commande de service

L'attribut de notification *Service Control Options List* donne une chaîne de bits désignant une ou plusieurs options de commande de service comme défini par le type de données ASN.1 **ServiceControlOptions** au § 7.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3.

```

serviceControlOptionsList ATTRIBUTE ::= {
    WITH SYNTAX    ServiceControlOptions
    SINGLE VALUE  TRUE
    ID            id-not-serviceControlOptionsList }

```

Lorsqu'un bit est activé dans le type **ServiceControlOptions**, il indique que la sélection d'option de commande de service correspondante est invalide, soit parce qu'on a demandé une option interdite ou non prise en charge, soit parce que l'option n'a pas été demandée lorsqu'elle était requise.

5.12.14 Localités à correspondances multiples

L'attribut de notification *Multiple Matching Localities* spécifie dans chaque valeur un ensemble d'assertions attributives qui, en cas d'application en fonction du répertoire toponymique, donneront une correspondance unique.

```

multipleMatchingLocalities ATTRIBUTE ::= {
    WITH SYNTAX    MultipleMatchingLocalities
    ID            id-not-multipleMatchingLocalities }

MultipleMatchingLocalities ::= SEQUENCE {
    matchingRuleUsed    MATCHING-RULE.&id OPTIONAL,
    attributeList      SEQUENCE OF AttributeValueAssertion }

```

L'élément **matchingRuleUsed** est présent sur option. Il peut être utilisé pour indiquer la règle de correspondance par mappage qui a été utilisée.

Aucune règle de correspondance n'est définie pour cet attribut. Plusieurs valeurs identiques ou presque identiques sont tolérées.

5.12.15 Elargissement proposé

L'attribut de notification *Proposed Relaxation* donne la séquence des éléments **MRMapping** qui peuvent être fournis dans le cadre de la politique d'élargissement fournie dans le composant **relaxation** d'une demande de recherche subséquente.

```

proposedRelaxation ATTRIBUTE ::= {
    WITH SYNTAX          MRMappings
    ID                  id-not-proposedRelaxation }

```

MRMappings ::= SEQUENCE OF MRMapping

La séquence de mappages **MRMapping** n'a pas de contenu.

5.12.16 Elargissement appliqué

L'attribut de notification *Applied Relaxation* est utilisé pour énumérer les attributs du filtre qui ont été soumis à un élargissement ou à un resserrement, autres que ceux qui ont été produits par l'élément **basic** d'une politique d'élargissement.

```

appliedRelaxation ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                  id-not-appliedRelaxation }

```

SECTION 3 – RÈGLES DE CONCORDANCE

6 Définition des règles de concordance

NOTE – Pour les définitions des règles **objectIdentifierMatch** et **distinguishedNameMatch**, voir la Rec. UIT-T X.501 | ISO/CEI 9594-2.

6.1 Règles de concordance par chaînes

Dans les règles de concordances spécifiées du § 6.1.1 au § 6.1.11, les espaces suivants ne sont pas considérés comme significatifs:

- les espaces de début (précédant le premier caractère qui n'est pas un espace);
- les espaces de fin (suivant le dernier caractère qui n'est pas un espace);
- les espaces internes consécutifs multiples (considérés comme équivalant à un seul caractère d'espace).

Une chaîne composée uniquement d'espaces est équivalente à une chaîne contenant exactement un seul espace.

Dans les règles de concordance auxquelles cette section est applicable, les chaînes à mettre en concordance le seront comme si les espaces non significatifs n'étaient pas présents dans les deux chaînes comparées.

6.1.1 Case Ignore Match (concordance sans exactitude de casse)

La règle *Case Ignore Match* vérifie l'égalité d'une chaîne présentée avec une valeur d'attribut de type **DirectoryString** ou un des types de données figurant dans le type de sélection **DirectoryString**, par exemple **UTF8String**, sans tenir compte de la casse (minuscules ou majuscules) des chaînes (par exemple, "Dundee" concordera avec "DUNDEE").

```
caselgnoreMatch MATCHING-RULE ::= {
  SYNTAX  DirectoryString {ub-match}
  ID      id-mr-caselgnoreMatch }
```

On obtient la valeur TRUE lorsque les chaînes ont la même longueur et que les caractères homologues sont identiques sauf, le cas échéant, en ce qui concerne la casse.

Si les chaînes à comparer ont une syntaxe abstraite numéro un différente, la comparaison s'effectuera normalement du moment que les caractères homologues se retrouvent dans les deux jeux de caractères. Sinon, la concordance échouera.

6.1.2 Case Ignore Ordering Match (concordance d'ordonnement sans exactitude de casse)

La règle *Case Ignore Ordering Match* compare l'ordre de fusionnement d'une chaîne présentée avec des valeurs d'attribut dont le type est **DirectoryString** ou un des types de données figurant dans le type de sélection **DirectoryString**, par exemple **UTF8String**, sans tenir compte de la casse (minuscules ou majuscules) des chaînes.

```
caselgnoreOrderingMatch MATCHING-RULE ::= {
  SYNTAX  DirectoryString {ub-match}
  ID      id-mr-caselgnoreOrderingMatch }
```

La règle renvoie la valeur TRUE si une quelconque valeur de l'attribut est "inférieure" ou apparaît plus tôt que la valeur présentée lorsque les chaînes sont comparées dans l'ordre de fusionnement normal pour leur syntaxe et que dans les deux chaînes les lettres minuscules ont été remplacées par leur équivalent en majuscules.

Si les chaînes à comparer ont une syntaxe abstraite numéro un différente, la comparaison s'effectuera normalement du moment que les caractères homologues se retrouvent dans les deux jeux de caractères. Sinon, la concordance échouera.

6.1.3 Case Ignore Substrings Match (concordance de sous-chaînes sans exactitude de casse)

La règle *Case Ignore Substrings Match* détermine si une valeur présentée est une sous-chaîne d'une quelconque valeur d'un attribut dont le type est **DirectoryString** ou un des types de données figurant dans le type de sélection **DirectoryString**, par exemple **UTF8String**, sans tenir compte de la casse (minuscules ou majuscules) des chaînes.

```

caseIgnoreSubstringsMatch MATCHING-RULE ::= {
  SYNTAX   SubstringAssertion
  ID       id-mr-caseIgnoreSubstringsMatch }

```

```

SubstringAssertion ::= SEQUENCE OF CHOICE {
  initial   [0]   DirectoryString {ub-match},
  any       [1]   DirectoryString {ub-match},
  final     [2]   DirectoryString {ub-match},
  control   Attribute } -- Élément utilisé pour spécifier l'interprétation des élément suivants
  -- au plus un seul composant initial et un seul composant final

```

La règle renvoie la valeur TRUE lorsque la valeur de l'attribut est subdivisée de telle sorte que:

- les sous-chaînes spécifiées (**initial**, **any**, **final**) concordent avec différentes fractions de la valeur dans l'ordre de la séquence **strings**;
- la sous-chaîne **initial**, lorsqu'elle existe, concorde avec la première fraction de la valeur;
- la sous-chaîne **final**, lorsqu'elle existe, concorde avec la dernière fraction de la valeur;
- la sous-chaîne **any**, lorsqu'elle existe, concorde avec une fraction arbitraire quelconque de la valeur.
- l'élément **control** n'est pas utilisé pour les correspondances **caseIgnoreSubstringsMatch**, **telephoneNumberSubstringsMatch** ni pour aucune autre forme de correspondance de sous-chaînes pour lesquelles seuls des éléments de type initial, quelconque ou final sont utilisés dans l'algorithme de mise en correspondance. Si un élément **control** est rencontré, il n'est pas pris en compte. L'élément de commande n'est utilisé que pour les règles de correspondance spécifiant explicitement son utilisation dans l'algorithme de mise en correspondance. Une telle règle de correspondance peut également redéfinir la sémantique des sous-chaînes **initial**, **any** et **final**.

NOTE – La règle de correspondance **generalWordMatch** est un exemple d'une telle règle.

Il doit exister au plus une sous-chaîne **initial** et au plus une sous-chaîne **final** dans la séquence **strings**. Lorsqu'elle existe, la sous-chaîne **initial** doit être le premier élément de la séquence **strings**. Lorsqu'elle existe, la sous-chaîne **final** doit être le dernier élément de la séquence **strings**. Le nombre de sous-chaînes **any** présentes dans la séquence **strings** peut être nul.

Pour qu'une sous-chaîne concorde avec une fraction de la valeur de l'attribut, les caractères correspondants doivent être identiques, sauf en ce qui concerne la casse. Lorsque les chaînes à comparer n'ont pas la même syntaxe ASN.1, la comparaison s'effectue normalement aussi longtemps que les caractères homologues existent dans les deux jeux de caractères. Dans le cas contraire, la concordance n'aboutit pas.

6.1.4 Case Exact Match (concordance avec exactitude de casse)

La règle *Case Exact Match* vérifie l'égalité d'une chaîne présentée avec des valeurs d'attribut dont le type est **DirectoryString** ou un des types de données figurant dans le type de sélection **DirectoryString**, par exemple **UTF8String**.

```

caseExactMatch MATCHING-RULE ::= {
  SYNTAX   DirectoryString {ub-match}
  ID       id-mr-caseExactMatch }

```

Cette règle est identique à celle de **caseIgnoreMatch**, sauf que l'on tient compte de la hauteur de casse.

6.1.5 Case Exact Ordering Match (concordance d'ordonnement avec exactitude de casse)

La règle *Case Exact Ordering Match* compare l'ordre de fusionnement d'une chaîne présentée avec des valeurs d'attribut dont le type est **DirectoryString** ou un des types de données figurant dans le type de sélection **DirectoryString**, par exemple **UTF8String**.

```

caseExactOrderingMatch MATCHING-RULE ::= {
  SYNTAX   DirectoryString {ub-match}
  ID       id-mr-caseExactOrderingMatch }

```

Cette règle est identique à celle de **caseIgnoreOrderingMatch**, sauf que les lettres minuscules ne sont pas remplacées par des majuscules.

6.1.6 Case Exact Substrings Match (concordance de sous-chaînes avec exactitude de casse)

La règle *Case Exact Substrings Match* détermine si une valeur présentée est une sous-chaîne d'une valeur quelconque d'un attribut dont le type est **DirectoryString** ou un des types de données figurant dans le type de sélection **DirectoryString**, par exemple **UTF8String**.

```
caseExactSubstringsMatch MATCHING-RULE ::= {
    SYNTAX    SubstringAssertion          -- choix PrintableString seulement
    ID        id-mr-caseExactSubstringsMatch }
```

Cette règle est identique à celle de **caseIgnoreSubstringsMatch**, sauf que l'on tient compte de la hauteur de casse.

6.1.7 Numeric String Match (concordance de chaînes numériques)

La règle *Numeric String Match* vérifie l'égalité d'une chaîne numérique présentée avec des valeurs d'attribut de type **NumericString**.

```
numericStringMatch MATCHING-RULE ::= {
    SYNTAX    NumericString
    ID        id-mr-numericStringMatch }
```

Cette règle est identique à celle de **caseIgnoreMatch**, sauf que tous les caractères d'espace sont omis pendant la comparaison (la hauteur de casse n'intervient pas puisque les caractères sont numériques).

6.1.8 Numeric String Ordering Match (concordance d'ordonnement de chaînes numériques)

La règle *Numeric String Ordering Match* compare l'ordre de fusionnement d'une chaîne présentée avec des valeurs d'attribut de type **NumericString**.

```
numericStringOrderingMatch MATCHING-RULE ::= {
    SYNTAX    NumericString
    ID        id-mr-numericStringOrderingMatch }
```

Cette règle est identique à celle de **caseIgnoreOrderingMatch**, sauf que tous les caractères d'espace sont omis pendant la comparaison (la hauteur de casse n'intervient pas puisque les caractères sont numériques).

6.1.9 Numeric String Substrings Match (concordance de sous-chaînes de chaînes numériques)

La règle *Numeric String Substrings Match* détermine si une valeur présentée est une sous-chaîne d'une valeur quelconque d'un attribut de type **NumericString**.

```
numericStringSubstringsMatch MATCHING-RULE ::= {
    SYNTAX    SubstringAssertion
    ID        id-mr-numericStringSubstringsMatch }
```

Cette règle est identique à celle de **caseIgnoreSubstringsMatch**, sauf que tous les caractères d'espace sont omis pendant la comparaison (la hauteur de casse n'intervient pas puisque les caractères sont numériques).

6.1.10 Case Ignore List Match (concordance de listes sans exactitude de casse)

La règle *Case Ignore List Match* vérifie l'égalité d'une séquence de chaînes présentée avec des valeurs d'attribut qui sont des séquences de **DirectoryStrings**, sans tenir compte de la casse (majuscules ou minuscules) de ces chaînes.

```
caseIgnoreListMatch MATCHING-RULE ::= {
    SYNTAX    CaseIgnoreList
    ID        id-mr-caseIgnoreListMatch }
```

```
CaseIgnoreList ::= SEQUENCE OF DirectoryString {ub-match}
```


Cette règle renvoie la valeur TRUE si et seulement si le nombre de chaînes est le même dans chaque séquence et que les chaînes correspondantes concordent. Cette dernière concordance suit la règle **caseIgnoreMatch**.

6.1.11 Case Ignore List Substrings Match (concordance de sous-chaînes de liste sans exactitude de casse)

La règle *Case Ignore List Substring* compare une sous-chaîne présentée avec des valeurs d'attribut qui sont des séquences de **DirectoryStrings**, mais sans tenir compte de la hauteur de casse (majuscules ou minuscules) lors des comparaisons.

```
caseIgnoreListSubstringsMatch MATCHING-RULE ::= {
  SYNTAX  SubstringAssertion
  ID      id-mr-caseIgnoreListSubstringsMatch }
```

Une valeur présentée concorde avec une valeur enregistrée si et seulement si la valeur présentée concorde avec la chaîne formée par concaténation des chaînes de la valeur enregistrée. Cette concordance est effectuée conformément à la règle **caseIgnoreSubstringsMatch**; aucune des valeurs **initial**, **any** ou **final** de la valeur présentée n'est toutefois prise en considération lors d'une comparaison avec une sous-chaîne de la chaîne concaténée dont la longueur dépasse celle d'une des chaînes de la valeur enregistrée.

6.1.12 Stored Prefix Match (concordance de préfixe enregistré)

La règle *Stored Prefix Match* détermine si une valeur d'attribut, dont la syntaxe est **DirectoryString**, est un préfixe, c'est-à-dire une sous-chaîne initiale, de la valeur présentée, sans tenir compte de la hauteur de casse (majuscules ou minuscules) des chaînes.

NOTE – Cette correspondance peut servir, par exemple, à comparer des valeurs qui, dans l'annuaire, sont des indicatifs de zones téléphoniques, avec une valeur prétendue qui est un numéro de téléphone.

```
storedPrefixMatch MATCHING-RULE ::= {
  SYNTAX  DirectoryString {ub-match}
  ID      id-mr-storedPrefixMatch }
```

La règle renvoie la valeur TRUE si la valeur de l'attribut est une sous-chaîne initiale de la valeur présentée telle que les caractères correspondants soient identiques bien que pouvant différer par la casse.

6.2 Règles de concordance à base syntaxique

6.2.1 Boolean Match (concordance de valeurs booléennes)

La règle *Boolean Match* vérifie l'égalité d'une valeur booléenne présentée avec des valeurs d'attribut de type **BOOLEAN**.

```
booleanMatch MATCHING-RULE ::= {
  SYNTAX  BOOLEAN
  ID      id-mr-booleanMatch }
```

Cette règle renvoie la valeur TRUE si les deux valeurs sont en même temps TRUE ou FALSE.

6.2.2 Integer Match (concordance d'entiers)

La règle *Integer Match* vérifie l'égalité d'une valeur entière ou énumérée présentée avec des valeurs d'attribut de type **INTEGER** ou **ENUMERATED** respectivement.

```
integerMatch MATCHING-RULE ::= {
  SYNTAX  INTEGER
  ID      id-mr-integerMatch }
```

Cette règle renvoie la valeur TRUE si la valeur entière ou énumérée présentée est égale à la valeur enregistrée.

6.2.3 Integer Ordering Match (concordance d'ordonnement d'entiers)

La règle *Integer Ordering Match* compare une valeur d'entier présentée avec des valeurs d'attribut de type **INTEGER**.

```
integerOrderingMatch MATCHING-RULE ::= {
    SYNTAX    INTEGER
    ID        id-mr-integerOrderingMatch }
```

Cette règle renvoie la valeur TRUE si une quelconque valeur de l'attribut est inférieure à la valeur présentée.

6.2.4 Bit String Match (concordance de chaînes de bits)

La règle *Bit String Match* compare une chaîne de bits présentée avec des valeurs d'attribut de type **BIT STRING**.

```
bitStringMatch MATCHING-RULE ::= {
    SYNTAX    BIT STRING
    ID        id-mr-bitStringMatch }
```

Cette règle renvoie la valeur TRUE lorsque la valeur d'attribut a le même nombre de bits pour la valeur présentée, et que les bits concordent d'homologue à homologue.

6.2.5 Octet String Match (concordance de chaînes d'octets)

La règle *Octet String Match* vérifie l'égalité d'une chaîne d'octets présentée avec des valeurs d'attribut de type **OCTET STRING**.

```
octetStringMatch MATCHING-RULE ::= {
    SYNTAX    OCTET STRING
    ID        id-mr-octetStringMatch }
```

Cette règle renvoie la valeur TRUE si et seulement si les chaînes ont la même longueur et si les octets correspondants sont identiques.

6.2.6 Octet String Ordering Match (concordance d'ordonnement de chaînes d'octets)

La règle *Octet String Ordering Match* compare l'ordre de fusionnement d'une chaîne d'octets présentée avec des valeurs d'attribut de type **OCTET STRING**.

```
octetStringOrderingMatch MATCHING-RULE ::= {
    SYNTAX    OCTET STRING
    ID        id-mr-octetStringOrderingMatch }
```

Cette règle compare des chaînes d'octets depuis le premier octet jusqu'au dernier et depuis l'élément binaire le plus significatif jusqu'à l'élément binaire le moins significatif de chaque octet. La première détection d'un bit différent détermine l'ordonnement des chaînes. Un bit zéro précède un bit un. Si les chaînes sont identiques mais contiennent différents nombres d'octets, la chaîne la plus courte précède la chaîne la plus longue.

6.2.7 Octet String Substrings Match (concordance entre sous-chaînes de chaînes d'octets)

La règle *Octet String Substrings Match* détermine si une chaîne d'octets présentée est une sous-chaîne de valeurs d'attribut de type **OCTET STRING**.

```
octetStringSubstringsMatch MATCHING-RULE ::= {
    SYNTAX    OctetSubstringAssertion
    ID        id-mr-octetStringSubstringsMatch }
```

```
OctetSubstringAssertion ::= SEQUENCE OF CHOICE {
    initial    [0]  OCTET STRING,
    any        [1]  OCTET STRING,
    final      [2]  OCTET STRING }
-- au plus un composant initial et un composant final
```

Comme la règle **caselgnoreSubstringsMatch**, cette règle renvoie la valeur TRUE lorsque la valeur d'attribut contient la séquence d'octets de la chaîne présentée.

6.2.8 Telephone Number Match (concordance de numéros téléphoniques)

La règle *Telephone Number Match* vérifie l'égalité d'une valeur présentée avec des valeurs d'attribut de type **PrintableString** qui sont des numéros de téléphone.

```
telephoneNumberMatch MATCHING-RULE ::= {
  SYNTAX  PrintableString
  ID      id-mr-telephoneNumberMatch }
```

Les règles de concordance sont les mêmes que pour **caselgnoreMatch**, sauf que tous les caractères d'espace et de tiret sont omis pendant la comparaison.

6.2.9 Telephone Number Substrings Match (concordance de sous-chaînes de numéros téléphoniques)

La règle *Telephone Number Substrings Match* détermine si une sous-chaîne présentée est une sous-chaîne de valeurs d'attribut de type **PrintableString** qui sont des numéros de téléphone.

```
telephoneNumberSubstringsMatch MATCHING-RULE ::= {
  SYNTAX  SubstringAssertion
  ID      id-mr-telephoneNumberSubstringsMatch }
```

Les règles de concordance sont les mêmes que pour **caseExactSubstringsMatch**, sauf que tous les caractères d'espace et de tiret sont omis pendant la comparaison.

6.2.10 Presentation Address Match (concordance d'adresses de présentation)

La règle *Presentation Address Match* vérifie l'égalité de valeurs présentées avec des valeurs d'attribut de type **PresentationAddress**.

```
presentationAddressMatch MATCHING-RULE ::= {
  SYNTAX  PresentationAddress
  ID      id-mr-presentationAddressMatch }
```

La règle renvoie la valeur TRUE si et seulement si les sélecteurs de l'adresse de présentation, présentés et enregistrés, sont égaux et si les **nAddresses** présentées sont un sous-ensemble de celles qui sont enregistrées.

6.2.11 Unique Member Match (concordance de membre unique)

La règle *Unique Member Match* vérifie l'égalité d'une valeur de membre unique présentée et d'une valeur d'attribut de type **NameAndOptionalUID**.

```
uniqueMemberMatch MATCHING-RULE ::= {
  SYNTAX  NameAndOptionalUID
  ID      id-mr-uniqueMemberMatch }
```

La règle renvoie la valeur TRUE si et seulement si les composants **dn** de la valeur d'attribut et de la valeur présentée concordent selon la règle **distinguishedNameMatch** et si l'élément **uid** n'est pas présent dans la valeur d'attribut ou concorde avec l'élément correspondant de la valeur présentée selon la règle **bitStringMatch**.

6.2.12 Protocol Information Match (concordance d'information de protocole)

La règle *Protocol Information Match* vérifie l'égalité de valeurs de **ProtocolInformation** présentées et de valeurs de même type.

```
protocolInformationMatch MATCHING-RULE ::= {
  SYNTAX  OCTET STRING
  ID      id-mr-protocolInformationMatch }
```

Les valeurs de syntaxe d'assertion sont dérivées des valeurs de syntaxe d'attribut par utilisation de l'élément **nAddress**.

La règle renvoie la valeur TRUE lorsque la valeur présentée et l'élément **nAddress** de la valeur enregistrée concordent selon la règle **octetStringMatch**.

6.3 Time matching rules (règles de concordance à base temporelle)

6.3.1 UTC Time Match (concordance d'UTC)

La règle *UTC Time Match* vérifie l'égalité d'une valeur présentée et d'une valeur d'attribut de type **UTCTime**.

```
uTCTimeMatch MATCHING-RULE ::= {
  SYNTAX   UTCTime
  ID       id-mr-uTCTimeMatch }
```

La règle renvoie la valeur TRUE si la valeur de l'attribut représente la même heure que la valeur présentée. Lorsque l'heure UTC donnée ne précise pas les secondes, le nombre de secondes est supposé égal à zéro.

6.3.2 UTC Time Ordering Match (ordonnancement d'après l'heure UTC)

La règle *UTC Time Ordering Match* compare la position temporelle d'une valeur présentée à la valeur d'attribut de type **UTCTime**.

```
uTCTimeOrderingMatch MATCHING-RULE ::= {
  SYNTAX   UTCTime
  ID       id-mr-uTCTimeOrderingMatch }
```

La règle renvoie la valeur TRUE lorsque la valeur d'attribut représente une heure antérieure à l'heure présentée. Les temps UTC dont les valeurs d'année vont de 50 à 99 doivent être considérés comme représentant des temps antérieurs à ceux dont les valeurs d'année vont de 00 à 49. Lorsque l'heure UTC ne précise pas les secondes, le nombre de secondes est supposé égal à zéro.

La valeur du champ "année" en deux chiffres sera traduite en un champ à quatre chiffres de la manière suivante:

- si la valeur est comprise entre 0 et 49 inclusivement, on lui rajoutera 2000;
- si la valeur est comprise entre 50 et 99 inclusivement, on lui rajoutera 1900.

6.3.3 Generalized Time Match (concordance d'heure généralisée)

La règle *Generalized Time Match* vérifie l'égalité d'une valeur présentée et d'une valeur d'attribut de type **GeneralizedTime** [selon le § 41.3 b) ou c) de la Rec. UIT-T X.680 | ISO/CEI 8824-1].

```
generalizedTimeMatch MATCHING-RULE ::= {
  SYNTAX   GeneralizedTime
           -- selon le § 41.3 b) ou c) de la Rec. UIT-T X.680 | ISO/CEI 8824-1
  ID       id-mr-generalizedTimeMatch }
```

La règle renvoie la valeur TRUE lorsque la valeur d'attribut représente la même heure que la valeur présentée. Lorsque l'heure spécifiée ne précise pas les minutes, ou les secondes, le nombre de minutes ou de secondes est supposé égal à zéro.

6.3.4 Generalized Time Ordering Match (ordonnancement selon l'heure généralisée)

La règle *Generalized Time Ordering Match* compare l'ordonnancement temporel d'une valeur présentée et d'une valeur d'attribut de type **GeneralizedTime** [selon le § 41.3 b) ou c) de la Rec. UIT-T X.680 | ISO/CEI 8824-1].

```
generalizedTimeOrderingMatch MATCHING-RULE ::= {
  SYNTAX   GeneralizedTime
           -- selon le § 41.3 b) ou c) de la Rec. UIT-T X.680 | ISO/CEI 8824-1
  ID       id-mr-generalizedTimeOrderingMatch }
```

La règle renvoie la valeur TRUE lorsque la valeur d'attribut représente une heure antérieure à l'heure présentée. Lorsque l'heure spécifiée ne précise pas les minutes, ou les secondes, le nombre de minutes ou de secondes est supposé égal à zéro.

6.3.5 System Proposed Match (correspondance proposée par le système)

La règle *System Proposed Match* est une règle de correspondance fictive, définie comme suit:

```
systemProposedMatch MATCHING-RULE ::= {
  ID       id-mr-systemProposedMatch }
```

Cette règle de correspondance peut être incluse par un demandeur dans la politique d'élargissement d'une demande de recherche afin d'indiquer que l'annuaire doit déterminer la règle de correspondance à utiliser dans une substitution de règle de correspondance.

6.4 First component matching rules (règles de concordance de premier élément)

6.4.1 Integer First Component Match (concordance de nombre entier)

La règle *Integer First Component Match* vérifie l'égalité d'une valeur de nombre entier présentée et d'une valeur d'attribut de type **SEQUENCE** dont le premier élément, obligatoire, est de type **INTEGER**.

```
integerFirstComponentMatch MATCHING-RULE ::= {
  SYNTAX  INTEGER
  ID      id-mr-integerFirstComponentMatch }
```

La règle renvoie la valeur TRUE lorsque la valeur d'attribut comprend un premier élément dont la valeur est égale au nombre entier présenté.

Les valeurs de la syntaxe d'assertion sont dérivées des valeurs de la syntaxe d'attribut par utilisation de la valeur du premier élément de la **SEQUENCE**.

6.4.2 Object Identifier First Component Match (concordance d'identificateur d'objet)

La règle *Object Identifier First Component Match* vérifie l'égalité d'une valeur d'identificateur d'objet présentée et de valeurs d'attribut de type **SEQUENCE** dont le premier élément, obligatoire, est de type **OBJECT IDENTIFIER**.

```
objectIdentifierFirstComponentMatch MATCHING-RULE ::= {
  SYNTAX  OBJECT IDENTIFIER
  ID      id-mr-objectIdentifierFirstComponentMatch }
```

La règle renvoie la valeur TRUE lorsque la valeur d'attribut comprend un premier élément dont la valeur correspond à l'identificateur d'objet présenté, selon les règles **objectIdentifierMatch**.

Les valeurs de la syntaxe d'assertion sont dérivées des valeurs de la syntaxe d'attribut par utilisation de la valeur du premier élément de la **SEQUENCE**.

6.4.3 Directory String First Component Match (concordance de premier composant de chaîne d'annuaire)

La règle *Directory String First Component Match* vérifie l'égalité d'une valeur **DirectoryString** présentée et d'une valeur d'attribut de type **SEQUENCE** dont le premier élément, obligatoire, est de type **DirectoryString**.

```
directoryStringFirstComponentMatch MATCHING-RULE ::= {
  SYNTAX  DirectoryString {ub-directory-string-first-component-match}
  ID      id-mr-directoryStringFirstComponentMatch }
```

La règle renvoie la valeur TRUE lorsque la valeur d'attribut comprend un premier élément dont la valeur correspond à la chaîne **DirectoryString** présentée, selon les règles **caseIgnoreMatch**.

Les valeurs de la syntaxe d'assertion sont dérivées des valeurs de la syntaxe d'attribut par utilisation de la valeur du premier élément de la **SEQUENCE**.

6.5 Word matching rules (règles de concordance de mots)

6.5.1 Word Match (concordance de mots)

La règle *Word Match* compare une chaîne présentée aux mots d'une valeur d'attribut de type **DirectoryString**.

```
wordMatch MATCHING-RULE ::= {
  SYNTAX  DirectoryString {ub-match}
  ID      id-mr-wordMatch }
```

La règle renvoie la valeur TRUE lorsque le mot présenté concorde avec l'un des mots de la valeur d'attribut. La concordance de mots individuels se fait selon la règle **caseIgnoreMatch**. La définition précise d'un "mot" relève du domaine local.

6.5.2 Keyword Match (concordance de mots clés)

La règle *Keyword Match* compare une chaîne présentée aux mots clés d'une valeur d'attribut de type **DirectoryString**.

```
keywordMatch MATCHING-RULE ::= {
  SYNTAX   DirectoryString {ub-match}
  ID       id-mr-keywordMatch }
```

La règle renvoie la valeur TRUE lorsque la valeur présentée correspond avec l'un quelconque des *mots clés* de la valeur d'attribut. L'identification des mots clés d'une valeur d'attribut et l'exactitude de la concordance sont deux questions relevant du domaine local.

6.5.3 General Word Match (correspondance générale de mots)

La règle *General Word Match* compare des mots contenus dans une chaîne présentée avec les mots contenus dans une valeur d'attribut dont le type est **DirectoryString**. La règle de concordance peut également être utilisée pour les valeurs d'attributs d'un type qui spécifie explicitement que sa syntaxe est l'une des sélections **DirectoryString**.

```
generalWordMatch MATCHING-RULE ::= {
  SYNTAX   SubstringAssertion
  ID       id-mr-generalWordMatch }
```

Cette règle de correspondance se différencie d'une règle normale de correspondance de sous-chaînes par l'insertion d'attributs de commande avant ou entre les éléments **initial**, **any** ou **final**. S'il n'y a pas d'attributs de commande dans l'élément de filtrage, la mise en correspondance doit être effectuée comme pour la règle de correspondance **caseExactSubstringsMatch** avec la sémantique des éléments **initial**, **any** et **final** comme défini par cette règle de correspondance. Si toutefois la règle de correspondance par égalité (éventuelle) pour le type d'attribut soumis à la correspondance est **caseIgnoreMatch**, c'est la règle **caseIgnoreSubstringsMatch** qui doit être utilisée à la place de la précédente.

Quatre types d'attribut de commande sont définis pour la correspondance générale de mots (des limitations sur leur emplacement sont définis ci-dessous). Tout autre attribut de commande doit être ignoré.

```
sequenceMatchType ATTRIBUTE ::= {
  WITH SYNTAX   SequenceMatchType
  SINGLE VALUE  TRUE
  ID            id-cat-sequenceMatchType } -- prenant la valeur par défaut sequenceExact
```

```
SequenceMatchType ::= ENUMERATED {
  sequenceExact           (0),
  sequenceDeletion       (1),
  sequenceRestrictedDeletion (2),
  sequencePermutation    (3),
  sequencePermutationAndDeletion (4),
  sequenceProviderDefined (5) }
```

```
wordMatchTypes ATTRIBUTE ::= {
  WITH SYNTAX   WordMatchTypes
  SINGLE VALUE  TRUE
  ID            id-cat-wordMatchType } -- prenant la valeur par défaut wordExact
```

```
WordMatchTypes ::= ENUMERATED {
  wordExact           (0),
  wordTruncated      (1),
  wordPhonetic       (2),
  wordProviderDefined (3) }
```

```
characterMatchTypes ATTRIBUTE ::= {
  WITH SYNTAX   CharacterMatchTypes
  SINGLE VALUE  TRUE
  ID            id-cat-characterMatchTypes }
```

```
CharacterMatchTypes ::= ENUMERATED {
    characterExact          (0),
    characterCaseIgnore     (1),
    characterMapped         (2) }
```

```
selectedContexts ATTRIBUTE ::= {
    WITH SYNTAX    ContextAssertion
    ID             id-cat-selectedContexts }
```

Chaque attribut affecte tous les éléments **initial**, **any** ou **final** suivants et les valeurs qu'il fournit remplacent celles qui étaient applicables jusque là.

Avant le premier attribut **sequenceMatchType** éventuel, la valeur qui doit être considérée comme applicable à cet attribut doit être choisie à **sequenceExact**. L'attribut n'a pas d'incidence sur l'évaluation des éléments **initial** et **final**, qui doivent toujours être interprétés comme une correspondance avec le mot initial et avec le mot final. Il n'a d'incidence que sur les mots qui restent sans correspondance. Le mot **initial**, s'il est présent, doit correspondre au premier mot du texte mémorisé. Si ces deux mots ne sont pas significatifs, ils doivent être considérés comme correspondants. L'emplacement des attributs **sequenceMatchType** définit les mots auxquels s'applique la forme de correspondance.

NOTE 1 – Pour de nombreux objectifs pratiques, il suffira de placer l'attribut **sequenceMatchType** avant le premier élément **initial**; des implémentations particulières pourront ne pas prendre en charge la totalité de cette définition.

Avant le premier attribut **wordMatchType** éventuel, la valeur qui doit être considérée comme applicable à cet attribut doit être choisie à **wordExact**. Avant le premier attribut **characterMatchType** éventuel, la valeur qui doit être considérée comme applicable à cet attribut doit être choisie à **characterExact**. Si cependant la règle (éventuelle) de correspondance par égalité pour le type d'attribut soumis à la correspondance est **caseIgnoreMatch**, c'est la règle **characterCaseIgnore** qui doit être utilisée à la place de la précédente.

Si l'attribut de commande **selectedContexts** est présent, il doit être le premier élément. Il ne doit y avoir qu'un seul de ces attributs de commande, qui doit être choisi comme une limitation de la valeur mémorisée (voir ci-dessous).

La règle renvoie la valeur TRUE si la valeur présentée contient une séquence non vide de mots correspondant au mot initial et au mot final spécifiés, ainsi que la séquence des mots restant sans correspondance dans la valeur d'attribut conformément au type **sequenceMatchType** spécifié, dans laquelle les mots correspondants sont adaptés en fonction des types **wordMatchTypes** spécifiés et dans laquelle les caractères correspondants dans les mots sont adaptés en fonction des types **characterMatchTypes**, sauf que si le composant **selectedContexts** est présent dans la valeur présentée, tous les éléments **ContextAssertion** sont également requis pour donner la valeur TRUE (comme spécifié dans la Rec. UIT-T X.501 | ISO/CEI 9594-2). La règle renvoie la valeur FALSE pour un certain attribut mémorisé si les mots ne correspondent pas ou si certains éléments **ContextAssertion** ne correspondent pas.

Un mot est une séquence non vide de caractères autres que d'espace, limitée par le début ou la fin de la chaîne ou par des caractères d'espace ou de ponctuation. Les caractères de ponctuation sont définis comme n'ayant pas d'incidence sur la sémantique des jetons: il s'agira normalement de virgules, guillemets, points finals de phrase, parenthèses, etc. La détermination de la nature des caractères de ponctuation relève d'une décision locale.

NOTE 2 – Par exemple, le caractère "!" est parfois utilisé dans un texte pour indiquer un son clappé comme il en existe dans certaines langues africaines. Il fait donc parfois partie d'un mot au lieu d'être un point d'exclamation (qui est un caractère de ponctuation).

De même, le mot désigné par l'élément **final**, s'il est présent, doit correspondre au dernier mot du texte mémorisé; si les deux mots ne sont pas significatifs, ils doivent être considérés comme correspondants.

Les mots non significatifs, qui sont des mots qui correspondent à l'un des mots d'une liste définie par l'implémentation de mots à poids sémantique faible (par exemple des articles et des prépositions) conformément aux types spécifiés par l'élément **characterMatchTypes** sont rejetés de la séquence de mots avant la mise en correspondance, sauf pour faire correspondre les mots désignés par les éléments **initial** et **final**, et la règle correspondante dans l'élément **wordMatchTypes** est rejetée de la séquence de règles si elle n'est pas la dernière de celles-ci.

La séquence de mots contenue dans la valeur présentée correspond à la séquence de mots contenue dans la valeur d'attribut, si celle-ci peut être transformée, conformément au type spécifié par l'élément **sequenceMatchType**, en une séquence contenant le même nombre de mots que la première séquence et ayant des mots correspondants adaptés. Si le type **sequenceMatchType** est **sequenceExact**, la transformation laisse la séquence inchangée. Si le type est **sequenceDeletion**, la transformation supprime zéro ou plus de zéro mots de la séquence. Si le type est **sequenceRestrictedDeletion**, la transformation supprime zéro ou plus de zéro mots sauf le premier mot de la séquence. Si le type est **sequencePermutation**, la transformation permute zéro ou plus de zéro mots de la séquence. Si le type est **sequencePermutationAndDeletion**, elle supprime zéro ou plus de zéro mots dans la séquence et permute zéro ou plus de zéro des mots restants. Si le type est **sequenceProviderDefined**, la transformation permute ou insère des mots en fonction d'une règle définie par la réalisation.

Un mot contenu dans la valeur présentée correspond à un mot dans la valeur d'attribut si ce dernier mot peut être transformé, conformément à la règle correspondante extraite des types **wordMatchTypes** spécifiés, en une séquence de caractères correspondant à leur tour aux caractères du mot contenu dans la valeur présentée. Chaque mot est mis en correspondance au moyen de la règle correspondante des types **wordMatchTypes**, dans lesquels la correspondance est déterminée avant d'appliquer de quelconques suppressions ou permutations à partir de la correspondance de séquence. Les éventuels mots dont le nombre dépasse celui des règles contenues dans l'élément **wordMatchTypes** sont mis en correspondance au moyen de la dernière règle. Si la règle est une correspondance exacte, la transformation laisse le mot inchangé. Si la règle est **wordTruncated**, zéro ou plus de zéro caractères sont enlevés de la fin du mot, jusqu'à obtenir une longueur de mot minimale, définie par l'implémentation. Si la règle est **wordPhonetic**, le mot est remplacé par un mot qui lui correspond selon un algorithme de correspondance phonétique qui est défini par la réalisation. S'il s'agit de la règle **wordProviderDefined**, le mot est mis en correspondance conformément à une règle définie par l'implémentation.

Les caractères contenus dans chaque mot sont comparés au moyen de la règle correspondante contenue dans l'élément **characterMatchTypes**, par laquelle la correspondance est déterminée avant d'appliquer de quelconques suppressions ou permutations à partir de la correspondance de séquence. Les caractères dont le nombre dépasse celui des règles contenues dans l'élément **characterMatchTypes** sont mis en correspondance au moyen de la dernière règle. Si le type contenu dans l'élément **characterMatchTypes** est **characterExact**, les caractères correspondants des mots correspondent s'ils sont identiques. Si le type est **characterCaseIgnore**, les caractères correspondants des mots correspondent s'ils sont identiques quel que soit leur cas. Si le type est **characterMapped**, les caractères correspondent s'ils renvoient au même caractère selon une table de mappage définie par l'implémentation. Cette table doit être telle que les caractères nationaux énumérés dans la Figure A.2/T.51 puissent être mis en correspondance au moyen des seuls caractères A à Z et 0 à 9 contenus dans les valeurs présentées et que de brèves séquences de caractères puissent être mises en correspondance avec un seul caractère, par exemple la séquence ae avec la diphtongue a-e ou la séquence ue avec u-tréma.

6.6 Règles de correspondance approchée

6.6.1 Approximate String Match (correspondance approchée des chaînes)

La règle de *Approximate String Match* compare une valeur présentée avec une valeur d'attribut conformément à un algorithme de mise en correspondance approchée qui est défini localement (par exemple variations de prononciation, correspondance phonétique, etc.). Cet algorithme doit être le même que celui qui a été invoqué en réponse au traitement d'un élément de filtre de type **approximateMatch** (voir la Rec. UIT-T X.511 | ISO/CEI 9594-3).

```
approximateStringMatch MATCHING-RULE ::= {
    ID          id-mr-approximateStringMatch }
```

La syntaxe d'assertion pour cette règle de correspondance est la même que pour la règle de correspondance par égalité applicable à l'attribut auquel elle se rapporte. Si aucune règle de correspondance par égalité n'est définie pour l'attribut, une syntaxe d'assertion quelconque est autorisée mais la règle donnera toujours une valeur indéfinie.

6.7 Règles de correspondance spéciales

6.7.1 Ignore if Absent Match (correspondance par non-prise en compte de l'absence)

La règle de *Ignore if Absent Match* compare une valeur applicable à un objet ou à un attribut quelconque.

```
ignoreIfAbsentMatch MATCHING-RULE ::= {
    ID          id-mr-ignoreIfAbsentMatch }
```

La règle renvoie les valeurs comme suit:

- a) si l'attribut est absent, la règle renvoie la valeur TRUE;
- b) si l'attribut est présent, la règle renvoie la valeur indéfinie.

Cette correspondance ne peut être utilisée que comme règle de correspondance hiérarchiquement ascendante (parente). Elle est ensuite utilisée de concert avec une règle de correspondance qui fait correspondre les valeurs lorsque l'attribut est présent. Voir également le § 13.5.2 de la Rec. UIT-T X.501 | ISO/CEI 9594-2.

NOTE – A l'intérieur d'une zone administrative propre à un service, l'on peut obtenir le même effet en spécifiant un sous-composant **defaultValues** vide du profil de demande d'attribut approprié.

6.7.2 Null Match (correspondance néant)

La règle de *Null Match* compare une valeur applicable à un objet ou à un attribut quelconque, selon la règle spéciale

```

nullMatch MATCHING-RULE ::= {
  ID          id-mr-nullMatch }

```

La règle renvoie les valeurs comme suit:

- a) si l'élément de filtrage n'est pas inversé, la règle renvoie la valeur TRUE;
- b) si l'élément de filtrage est inversé, la règle renvoie la valeur FALSE.

Cette correspondance peut être utilisée formellement pour provoquer la non-prise en compte d'un élément de filtrage. Celui-ci est en effet considéré comme absent s'il utilise la correspondance néant lors de l'évaluation de la compatibilité avec des règles de recherche.

6.8 Zonal Match (correspondance zonale)

Une *Zonal Match* est essentiellement applicable aux demandes de recherche faisant appel à des éléments de filtrage mappables par des tables géographiques. De tels éléments de filtrage peuvent être des assertions pour les éléments **localityName**, **stateOrProvinceName**, **postalCode**, etc.

La correspondance zonale utilise des éléments de filtrage combinables pour l'adaptation en fonction de la table de mappage.

La correspondance zonale peut tenir compte du fait que la perception des localités par les utilisateurs peut différer du modèle toponymique utilisé dans un domaine DMD. Le mappage entre cette perception des utilisateurs et le modèle utilisé dans un domaine DMD doit tenir compte du fait qu'un utilisateur peut rechercher des localités qui ne sont pas directement reproduites dans des entrées de l'annuaire ou dans leur nom. De telles localités peuvent être floues en ce sens qu'elles ne se rapportent pas exactement à des localités qui sont plus officielles. De même, un utilisateur peut se tromper légèrement en choisissant un nom de localité au cours d'une recherche si la personne recherchée vit près de la limite avec une localité voisine. C'est pourquoi l'on subdivise une région, comme un pays, en *zones*. Celles-ci sont des surfaces entièrement contenues dans le domaine d'une localité citée en référence lors d'une demande de recherche. Le résultat d'un mappage des éléments de filtrage mappables est une liste de zones. On trouvera à l'Annexe E de plus amples explications sur la correspondance zonale.

En correspondance zonale, la table de mappage est appelée *répertoire toponymique* (c'est-à-dire qu'il s'agit d'un dictionnaire géographique). A l'intérieur du filtre, un ensemble d'éléments combinables de filtrage de localités peut permettre, en tant que tel, de définir un unique *lieu nommé* (c'est-à-dire une zone locale unique, habituellement contiguë). Ou bien, si cela est autorisé, de définir un petit nombre de lieux nommés qui correspondent aux éléments de filtrage. Un lieu nommé est un lieu du monde réel portant un nom de lieu distinct, comme une ville, un village, un canton, etc.

Un répertoire toponymique couvrira généralement (c'est-à-dire fournira une base de données géographiques relatives à) un domaine composé d'un seul pays ou d'une seule région. Lors d'une recherche géographique, une interrogation doit être interprétée dans les termes d'un répertoire toponymique spécifique. La façon dont le champ d'une recherche est déterminé et dont un répertoire toponymique approprié est sélectionné relève d'une décision locale mais la sélection peut être effectuée au moyen d'un répertoire toponymique par défaut pour l'agent DSA ou être fondée sur un ou plusieurs attributs (tels que **countryName**, **stateOrProvinceName** ou **localityName**) associés à l'opération de recherche (par exemple présents dans le cadre du nom distinctif de l'objet de base ou dans le cadre du filtre).

La première étape d'une correspondance zonale consiste à utiliser de concert un ou plusieurs éléments de filtrage pour identifier un ou plusieurs lieux nommés. L'on utilise à cette fin des éléments combinables de filtrage de localités (c'est-à-dire que tous les éléments de filtrage de localités sont contenus dans un même sous-filtre).

Autrement dit, la procédure identifie dans cette phase un ou plusieurs lieux nommés. A ce stade, il n'a été fait aucune référence que ce soit à des informations contenues dans l'arbre DIT. Le reste du filtre peut donc servir à identifier toutes les entrées relevant du champ de recherche qui possèdent des positions correspondant à ces lieux nommés, comme indiqué ci-dessous. Un élargissement peut être appliqué de façon que les lieux nommés correspondent à un plus grand nombre de positions d'entrée, afin d'éviter le renvoi de résultats inappropriés.

La correspondance zonale ne prend pas en charge le resserrement du champ de recherche.

Chaque entrée qui doit être considérée comme susceptible d'une mise en correspondance doit avoir une position identifiée soit par un lieu nommé unique, utilisant éventuellement plusieurs valeurs toponymiques (par exemple "Newton", "Chester", "Cheshire"), soit par une ou plusieurs zones (voir l'alinéa suivant) représentées par des valeurs placées dans un attribut de zone. Si une entrée possède des zones pour définir sa position, elle peut posséder également des valeurs de localité mais celles-ci seront, dans ce cas, informationnelles. L'autorité administrative est chargée de veiller à ce que les informations toponymiques correspondent à des lieux nommés réels.

Les zones sont des composants géographiques élémentaires non superposés, de nature distincte des lieux, de sorte qu'un lieu soit précisément composé d'une ou de plusieurs zones, comme énuméré dans le répertoire toponymique. Les zones sont désignées par des valeurs de chaîne qui sont uniques à l'intérieur d'une région toponymiquement répertoriée. Ainsi, deux lieux en intersection auront en commun une ou plusieurs zones correspondant à l'aire de leur intersection. Les zones sont représentées dans les entrées sous la forme d'attributs, éventuellement de type opérationnel. Dans ce cas, les informations zonales ne seront jamais renvoyées sous forme de valeurs attributives, à moins que l'attribut représentant la zone ne soit expressément demandé en tant qu'attribut opérationnel. En variante, une zone peut être représentée par un attribut normal (par exemple **postalCode**). Les valeurs toponymiques sont renvoyées comme d'habitude, sous réserve du contrôle d'accès.

NOTE 1 – La nature exacte d'une zone et son mappage sur un attribut spécifique relèvent d'une décision locale. Ils dépendront probablement des capacités d'une implémentation particulière. Au Royaume-Uni, un code postal comme "RG12 2JL" conviendrait bien pour désigner une zone car il définit souvent une petite aire comme un côté de rue. Dans les villes, les zones seront donc petites alors qu'elle seront d'autant plus grandes en campagne. Dans les régions sans occupation démographique ni constructions (comme les déserts), une zone pourra même être très grande.

En cas d'intersection entre zones définies pour un lieu nommé et zones définies pour une entrée (c'est-à-dire si une règle de correspondance fondée sur une intersection est utilisée), la position d'entrée (définie par ces zones) correspondra au lieu nommé défini dans le répertoire toponymique. Si la position d'entrée est définie comme étant un lieu nommé, on considérera que cette position se compose des zones constitutives de ce lieu nommé.

La correspondance zonale permet des correspondances étendues (c'est-à-dire élargies), où le niveau 0 correspond à la définition de base des objets énumérés dans le répertoire toponymique. Les niveaux 1 et au-dessus correspondent à un élargissement progressif et systématique des zones constitutives d'un lieu, de façon qu'un plus grand nombre d'emplacements correspondent.

On trouvera ci-dessous un énoncé plus formel du modèle sous-tendant la correspondance zonale:

- a) la correspondance zonale est fondée sur l'existence d'un ou de plusieurs *répertoires toponymiques* qui sont pris en charge à cette fin par des agents DSA. Un répertoire toponymique est un dictionnaire géographique dont le domaine recouvre un pays ou une *région* toponymique, au moyen d'une base de données appropriée. La sélection du domaine d'une recherche spécifique est effectuée par des moyens locaux. Le répertoire toponymique contient des toponymes avec leurs propriétés, y compris des listes de lieux nommés correspondants. Il est pris en charge par des mécanismes d'extraction et de collationnement des propriétés toponymiques indiquées par des attributs combinables de localité. Ce répertoire est tout à fait indépendant de l'arbre DIT;
- b) la région couverte par un répertoire toponymique contient des *lieux*. Chaque lieu est une aire géographique portant un nom distinctif; les lieux peuvent s'intersecter et peuvent même s'étendre un peu au-delà de la frontière d'une région. Les lieux qui sont identifiables par référence au répertoire toponymique sont appelés *lieux nommés*;
- c) le répertoire toponymique lui-même est fondé sur des chaînes qui sont des *toponymes*. Ceux-ci sont utilisés pour identifier (ou nommer) des lieux nommés. Le nom d'un lieu nommé peut être:
 - un simple toponyme, comportant éventuellement plusieurs mots;
 - un ensemble de toponymes correspondant généralement à une assez grande surface et englobant d'autres toponymes correspondant (dans le contexte) à une plus petite surface,
- d) le concept de surfaces assez grandes et plus petites peut parfois être représenté utilement dans la caractéristique d'échelle appliquée à un lieu. Exemples non formels de lieux à échelle variable: parcelles, lotissements, villages, villes, cités, cantons, départements, pays. En général, un lieu nommé devra être associé, dans le répertoire toponymique, aux noms des lieux l'englobant sur une plus grande échelle, même si ces lieux ne sont pas nécessaires pour une identification unique;
- e) les toponymes peuvent aussi avoir des synonymes associés à un lieu particulier, qui peuvent (par exemple) être des abréviations ou des variantes nominatives. Il est pratique de définir un nom canonique pour chaque lieu, sur lequel peuvent être mappés des synonymes des toponymes constitutifs;
- f) les toponymes peuvent parfois être déduits de toponymes plus simples par utilisation de composants sémantiques tels que "près de" (par exemple "Près de Tenterden"). Cela peut théoriquement servir à définir un lieu annulaire autour de la ville de Tenterden dans la région du Kent (Angleterre); mais cette désignation sera probablement mieux comprise comme un toponyme ne définissant pas de lieu précis;

- g) tous les lieux couverts par le répertoire toponymique doivent avoir un unique nom canonique composé d'un ensemble distinct de toponymes, ceux-ci pouvant être ordonnés selon l'échelle que chaque toponyme implique dans le contexte;
- h) les lieux sont subdivisés en zones telles qu'elles soient toujours imbriquées dans chaque lieu. Chaque partie d'un lieu possède une zone correspondante. Une zone est le bloc de construction des lieux dans un répertoire toponymique; chaque point d'une région est contenu dans une zone particulière;
- i) les zones possèdent habituellement des zones voisines (sauf blocage effectif par une limite géographique ou politique majeure telle qu'un lac, un fleuve, un bord de mer, une montagne ou une frontière nationale). La surface définie par un lieu pourra donc, le plus souvent, être étendue par inclusion de zones voisines de celles qui composent ce lieu. Cette extension pourra être effectuée indéfiniment par échelons successifs. L'inclusion d'un niveau donné d'extension par voisinage est appelée *extension-1* d'un lieu; le niveau d'extension suivant sera appelé *extension-2* et ainsi de suite. Le champ d'extension peut être réglable (par augmentation ou réduction) localement afin de représenter une situation concrète, mais de tels réglages devraient être relativement rares;
- j) une entrée représentant un objet physique peut être définie comme contenant un *emplacement*. Un emplacement peut se définir comme un ensemble de zones contenu dans un attribut de zone approprié ou comme un lieu nommé faisant appel à un ou plusieurs toponymes contenus dans un attribut de localité tel que **locationName**, cet attribut pouvant également être représenté comme un ensemble de zones. Une entrée correspondra à un lieu si l'ensemble des zones composant sa localité recouvre l'ensemble des zones qui représentent ce lieu (avec éventuelle extension-n) après consultation du répertoire toponymique comme décrit ci-dessus;
- k) la sélection de zones, de lieux, de toponymes et la compilation de leurs relations relèvent d'une décision locale;
- l) les entrées qui correspondent par égalité sur la base des chaînes qu'elles contiennent doivent continuer à correspondre (ce qui évite pratiquement la correspondance zonale).

Afin de mieux préciser la correspondance zonale, l'on définit la classe d'objets informationnels non générique **ZONAL-MATCHING** comme une spécialisation de la classe d'objets informationnels générique **MAPPING-BASED-MATCHING**. Une instance de cette classe d'objets informationnels détermine les caractéristiques de la correspondance zonale.

ZONAL-MATCHING ::= MAPPING-BASED-MATCHING { ZonalSelect, TRUE, ZonalResult, zonalMatch }

Une instance de cette classe d'objets informationnels se caractérise par ce qui suit:

- a) la référence fictive **&selectBy**, si présente, est remplacée, au moyen de cette classe d'objets informationnels, par un ensemble de types d'attribut. La sélection d'une instance de cette classe d'objets informationnels est fondée sur ces attributs et sur les types d'attribut représentés dans le filtre de recherche. Une instance d'objet informationnel peut être sélectionnée si tous les types d'attribut représentés par ce composant sont représentés dans le filtre. Les sous-types d'attribut ne sont pas considérés (c'est-à-dire que la sélection doit être fondée sur des attributs explicitement nommés). Cependant, des critères locaux non définis par la présente Spécification d'annuaire peuvent également être pris en considération pour sélectionner une instance. Par exemple, la sélection peut être partiellement déterminée par l'objet de base **baseObject** de l'argument de recherche. Si ce composant est absent, la sélection se fonde entièrement sur la prise d'une décision locale;
- b) le composant **&ApplicableTo** doit spécifier un ensemble de types d'attribut associés à des localités et déterminés par des exigences locales, comme **localityName**, **stateOrProvinceName**, **streetName**, **postalCode**, etc;
- c) le composant **&subtypeIncluded** est activé conformément aux exigences locales;
- d) la référence de valeur fictive **&combinable** est remplacée inconditionnellement par la valeur TRUE;
- e) la référence de type fictive **&mappingResults** est remplacée, au moyen de cette classe d'objets informationnels, par le type de données **ZonalResult**;
- f) le composant **&userControl** est activé conformément aux exigences locales;
NOTE 2 – Ce champ devrait, dans la plupart des cas, prendre la valeur **TRUE**.
- g) le composant **&exclusive** est activé conformément aux exigences locales.
NOTE 3 – Une instance d'objet informationnel de cette classe d'objets informationnels est candidate à un élargissement exclusif.

ISO/CEI 9594-6:2001 (F)

- h) le composant **&matching-rule** est mis, par cette classe d'objets informationnels dérivée, à la valeur **zonalMatch**;
- i) le composant **&id** donne une identification unique de l'instance de l'algorithme de correspondance zonale.

Le type de données **ZonalSelect** est le suivant:

ZonalSelect ::= SEQUENCE OF AttributeType

Le type de données **ZonalResult** est utilisé pour indiquer des conditions exceptionnelles pour la correspondance zonale.

ZonalResult ::= ENUMERATED {
 cannot-select-mapping (0),
 zero-mappings (2),
 multiple-mappings (3) }

La valeur

- a) **cannot-select-mapping** est le résultat lorsque les informations fournies dans le nom d'objet de base et dans le sous-filtre sont insuffisantes pour identifier le mappage à utiliser dans la règle de correspondance zonale. La correspondance ainsi effectuée produit un résultat indéfini. Aucun des sous-filtres possédant des éléments de filtrage conformes à la spécification du composant **&applicableTo** ne donneront donc la valeur **TRUE**.

NOTE 4 – A l'intérieur d'une zone administrative propre à un service et pour des règles de recherche correctement conçues, l'analyse de l'argument de recherche devrait y avoir détecté des informations insuffisantes.

- b) **zero-mappings** est le résultat lorsque les informations fournies dans l'élément (les éléments) de filtrage en vue d'un mappage ne peuvent pas être mappées, soit parce que aucun élément correspondant n'existe dans la table de mappage soit parce que le processus de mappage a produit zéro élément de filtrage à mettre en correspondance avec des entrées. Dans cette situation, une erreur **serviceError** doit être renvoyée avec le problème **requestedServiceNotAvailable**. Le composant de notification des résultats **CommonResults** doit contenir:
 - i) un attribut de notification **searchServiceProblem** ayant pour valeur **id-pr-ambiguousKeyAttributes**;
 - ii) un attribut de notification **filterItem** indiquant les éléments de filtrage mappables qui ne sont pas en mesure de fournir une correspondance.
- c) **multiple-mappings** est le résultat lorsque les informations fournies dans l'élément (les éléments) de filtrage peuvent être correctement mappées sur des entrées multiples du répertoire toponymique. La correspondance ainsi obtenue produit la valeur **TRUE** mais peut aussi conduire à abandonner la recherche pour cause d'erreur. Dans cette situation, une erreur **serviceError** doit être renvoyée avec le problème **requestedServiceNotAvailable**. Le composant de notification des résultats **CommonResults** doit contenir:
 - i) un attribut de notification **searchServiceProblem** ayant pour valeur **id-pr-ambiguousKeyAttributes**;
 - ii) un attribut de notification **multipleMatchingLocalities** comme indiqué par la règle de correspondance **zonalMatch**.

La règle de correspondance par mappage **zonalMatch** est associée à toute instance de la classe d'objets informationnels **ZONAL-MATCHING** comme suit:

zonalMatch MATCHING-RULE ::= {
 UNIQUE-MATCH-INDICATOR multipleMatchingLocalities
 ID id-mr-zonalMatch }

Cette règle de correspondance par mappage comporte le champ **UNIQUE-MATCH-INDICATOR** qui implique que la mise en correspondance en fonction du répertoire toponymique doit donner un résultat non ambigu. Si plusieurs entrées de la table donnent une correspondance au cours du processus de mappage, l'attribut de notification **multipleMatchingLocalities** attribute (voir § 5.12.14) est renvoyé dans le paramètre **notification** des résultats **CommonResults** de la recherche **search** ainsi qu'une erreur d'attribut avec le problème **ambiguousKeyAttributes**. Une valeur de l'attribut de notification **multipleMatchingLocalities** est incluse pour chaque entrée de table trouvée en correspondance avec le répertoire toponymique. Chacune de ces valeurs doit être une spécification d'un ensemble d'assertions **AttributeValueAssertion** qui, s'il est fourni par des éléments de filtrage par égalité mis en relation par opérateur AND dans chaque sous-filtre, donnerait une correspondance unique en fonction de l'entrée de table correspondante. Cela permettra à l'utilisateur de sélectionner, lors d'une demande de recherche subséquente, l'une des valeurs d'attribut de notification qui ont été renvoyées afin d'en tenir compte dans le filtre.

SECTION 4 – CONTEXTES

7 Définition des types de contextes

La présente Spécification d'annuaire définit un certain nombre de types de contextes pouvant être utiles dans une large gamme d'applications de l'annuaire.

7.1 Language Context (contexte linguistique)

Le type de contexte *Language Context* associe une valeur d'attribut à un ou plusieurs langages particuliers:

```
languageContext CONTEXT ::= {
  WITH SYNTAX   LanguageContextSyntax
  ID            id-avc-language }
```

LanguageContextSyntax ::= PrintableString (SIZE(2..3)) -- Codes de ISO 639-2 seulement

Il est considéré qu'une valeur présentée concorde avec une valeur enregistrée si, et seulement si, la séquence de caractères figurant dans la valeur présentée est identique à celle de la valeur enregistrée.

7.2 Temporal Context (contexte temporel)

Le type de contexte *Temporal Context* associe une valeur d'attribut à un ensemble de moments. Des expressions variées des moments sont possibles, parmi lesquelles figurent:

- a) un moment absolu de démarrage ou d'arrêt, par exemple le 14 décembre 1994 à 24h00;
- b) des tranches horaires données au cours de la journée, par exemple de 9h00 à 17h00;
- c) des jours de la semaine, par exemple lundi;
- d) des jours du mois, par exemple le 10, l'avant-dernier jour, etc.;
- e) des mois de l'année, par exemple mars;
- f) une année donnée, par exemple 1995;
- g) des semaines du mois, par exemple la deuxième semaine;
- h) un jour ou une semaine répétitif, par exemple la deuxième semaine de chaque mois;
- i) des négations logiques, par exemple pas le lundi.

```
temporalContext CONTEXT ::= {
  WITH SYNTAX   TimeSpecification
  ASSERTED AS   TimeAssertion
  ID            id-avc-temporal }
```

```
TimeSpecification ::= SEQUENCE {
  time          CHOICE {
    absolute    SEQUENCE {
      startTime [0] GeneralizedTime OPTIONAL,
      endTime   [1] GeneralizedTime OPTIONAL },
    periodic    SET OF Period },
  notThisTime  BOOLEAN DEFAULT FALSE,
  timeZone     TimeZone OPTIONAL }
```

```
Period ::= SEQUENCE {
  timesOfDay [0] SET SIZE (1..MAX) OF DayTimeBand OPTIONAL,
  days       [1] CHOICE {
    intDay   SET OF INTEGER,
    bitDay   BIT STRING { sunday (0), monday (1), tuesday (2),
      wednesday (3), thursday (4), friday (5), saturday (6) },
    dayOf    XDayOf } OPTIONAL,
  weeks     [2] CHOICE {
    allWeeks NULL,
    intWeek  SET OF INTEGER,
    bitWeek  BIT STRING { week1 (0), week2 (1), week3 (2), week4 (3),
      week5 (4) } } OPTIONAL,
```

```

months      [3] CHOICE {
                allMonths    NULL,
                intMonth     SET OF INTEGER,
                bitMonth     BIT STRING { january (0), february (1), march (2), april (3),
                                may (4), june (5), july (6), august (7), september (8),
                                october (9), november (10), december (11) }
            } OPTIONAL,
years       [4] SET OF INTEGER (1000 .. MAX) OPTIONAL }

```

```

XDayOf ::= CHOICE {
    first      [1] NamedDay,
    second    [2] NamedDay,
    third     [3] NamedDay,
    fourth    [4] NamedDay,
    fifth     [5] NamedDay }

```

```

NamedDay ::= CHOICE {
    intNamedDays  ENUMERATED {
        sunday      (1),
        monday      (2),
        tuesday     (3),
        wednesday   (4),
        thursday    (5),
        friday       (6),
        saturday    (7) },
    bitNamedDays  BIT STRING { sunday (0), monday (1), tuesday (2),
                                wednesday (3), thursday (4), friday (5), saturday (6) } }

```

```

DayTimeBand ::= SEQUENCE {
    startDayTime [0] DayTime DEFAULT { hour 0 },
    endDayTime   [1] DayTime DEFAULT { hour 23, minute 59, second 59 } }

```

```

DayTime ::= SEQUENCE {
    hour   [0] INTEGER (0..23),
    minute [1] INTEGER (0..59) DEFAULT 0,
    second [2] INTEGER (0..59) DEFAULT 0 }

```

```

TimeZone ::= INTEGER (-12..12)

```

```

TimeAssertion ::= CHOICE {
    now      NULL,
    at      GeneralizedTime,
    between SEQUENCE {
        startTime [0] GeneralizedTime,
        endTime   [1] GeneralizedTime OPTIONAL,
        entirely  BOOLEAN DEFAULT FALSE } }

```

Faire pour l'élément **time (temps)** le choix **absolute (absolu)** permet d'exprimer un instant particulier ou une tranche horaire au moyen de notations temporelles absolues sous la forme **GeneralizedTime** (temps généralisé universel). On exprime un instant particulier en assignant à **startTime (heure de début)** la même valeur qu'à **endTime (heure de fin)**. Si par contre **startTime** et **endTime** prennent des valeurs différentes, on a alors exprimé une durée. Si **endTime** manque, on a indiqué tout le temps à partir de **startTime**.

periodic (périodique) permet de spécifier **time** sous forme d'un ensemble de plages temporelles, l'effet combiné étant celui d'un OU logique sur l'ensemble.

NOTE 1 – Une autre solution consisterait à associer au contexte temporel une valeur d'attribut munie de plusieurs valeurs de contexte, une par période, puisque le résultat en serait aussi un OU logique. Le choix de SET OF a cependant été fait ici afin de permettre à **notThisTime (pas ce moment)** d'agir sur l'ensemble et de créer ainsi un "ni" logique. Lorsque **notThisTime** a la valeur FALSE, le choix de l'approche à suivre pour spécifier une période de temps est laissé au concepteur de la spécification.

Au sein de chaque **Period (période)**, chaque élément de SEQUENCE OF est considéré comme "inclus" dans l'élément suivant de SEQUENCE OF. SEQUENCE OF se présente dans un ordre croissant de l'unité temporelle, bien que la présence de tous ses niveaux ne soit pas nécessaire.

L'élément final de **Period** est supposé valide pour toutes les périodes d'unité temporelle supérieure.

NOTE 2 – Si par exemple une **Period SEQUENCE OF** se termine par **timesOfDay**, cet élément est tenu pour applicable tous les jours.

timesOfDay (moments du jour) indique les tranches horaires valables au cours des jours spécifiés dans l'élément suivant de **Period**. Si **days (jours)** n'est pas l'élément suivant, les tranches horaires sont valables pour tous les jours possibles de l'élément suivant. Si **timesOfDay** n'est pas présent, tous les moments de la journée sont valides dans l'élément suivant. Il est permis de spécifier des tranches horaires différentes pour des jours différents par usage d'occurrences multiples de **Period**.

L'élément **days** exprime des jours particuliers de la semaine, du mois ou de l'année, selon la nature de l'élément suivant de **Period**. Si **days** précède **weeks (semaines)** dans **Period**, il exprime alors des jours de la semaine et les valeurs de INTEGER sont restreintes au champ 1 à 7, où 1 représente dimanche. Si **days** précède **months (mois)** dans **Period**, il exprime alors des jours du mois et les valeurs de INTEGER sont restreintes au champ 1 à 31, où 1 représente le premier jour du mois. Si **days** précède **years (années)** dans **Period**, il exprime alors des jours de l'année et les valeurs de INTEGER sont restreintes au champ 1 à 366, où 1 représente le premier jour de l'année.

dayOf (journée de) sert à indiquer la première, seconde, troisième, quatrième et cinquième occurrence de **NamedDay (jour nommé)** dans le mois comme, par exemple, le premier lundi du mois ou les seconds mardi et vendredi d'août. L'emploi de **fifth (cinquième)** doit toujours indiquer le dernier **NamedDay** du mois considéré, comme par exemple le dernier mardi de juillet. S'il est fait pour **days** le choix de **dayOf**, alors l'élément **weeks** de **Period** est dépourvu de signification. Il est ignoré.

Si **days** n'est pas spécifié, alors tous les jours de la semaine sont valides au sein de l'élément suivant de **Period**.

L'élément **weeks** exprime des semaines précises du mois ou de l'année, selon l'élément suivant de **Period**. Si **weeks** précède **months** dans **Period**, il exprime alors des semaines du mois et les valeurs de INTEGER sont restreintes au champ 1 à 5, où 1 représente la première semaine du mois. Il est impératif d'admettre que la première semaine du mois soit la première semaine d'au moins quatre jours de ce mois. La cinquième semaine est toujours la dernière semaine du mois.

Si **weeks** précède **years** dans **Period**, il exprime alors des semaines de l'année et les valeurs de INTEGER sont restreintes au champ 1 à 53, où 1 représente la première semaine de l'année. Il est impératif d'admettre que la première semaine du mois soit la première semaine d'au moins quatre jours de cette année. La 53^e semaine est toujours la dernière semaine de l'année.

Si **allWeeks (toutes les semaines)** est spécifié, alors toutes les semaines sont valides au sein de l'élément suivant de **Period**. Par ce moyen, **days** peut exprimer des jours de toutes les semaines.

Si **weeks** n'est pas spécifié, alors toutes les semaines sont valides au sein de l'élément suivant de **Period**.

L'élément **months** exprime des mois donnés de l'année. Lorsqu'il est exprimé au moyen de valeurs de la forme INTEGER, ces valeurs sont restreintes au champ 1 à 12, où 1 est le premier mois de l'année (par exemple, janvier).

Si **allMonths (tous les mois)** est spécifié, alors tous les mois de l'année sont valides. Par ce moyen, **weeks** peut exprimer des semaines de tous les mois ou, si **weeks** n'est pas spécifié, **days** peut exprimer des jours de tous les mois.

Si **months** n'est pas spécifié, alors tous les mois de l'année sont valides.

Le composant **years (années)** exprime une ou plusieurs années. Si **years** n'est pas spécifié, alors toutes les années sont valides.

timeZone (zone horaire) exprime le fuseau horaire en heures de décalage par rapport à l'heure GMT dans laquelle est exprimé **time**. En l'absence de **timeZone**, un DSA qui traite le contexte temporel doit interpréter **time** par référence à son propre fuseau horaire.

Si **notThisTime** a la valeur **FALSE**, alors la valeur du contexte temporel est le moment exprimé par **time** dans **TimeSpecification (spécification temporelle)**. Si **notThisTime** a la valeur **TRUE**, alors la valeur du contexte temporel correspond à tout le temps qui n'est pas celui qu'exprime **time** dans **TimeSpecification**, réalisant ainsi une opération logique NOT.

On considère qu'il y a accord entre une assertion temporelle et une spécification temporelle s'il existe un recouvrement entre les périodes de temps considérées. Si l'assertion temporelle contient **now (maintenant)**, alors l'heure actuelle est utilisée dans l'évaluation. Si **now** ou bien **at (à)** est spécifié, alors l'assertion est considérée comme vraie si le moment précis tombe parmi les moments prévus dans la **TimeSpecification** enregistrée. Si l'assertion temporelle utilise **between (entre)** et si **entirely** a la valeur **FALSE**, alors l'assertion est considérée comme vraie si une portion quelconque de la période **between** tombe parmi les moments prévus dans la **TimeSpecification** enregistrée (il n'est pas nécessaire que le recouvrement soit total, il suffit qu'il y ait une période de recouvrement entre les deux spécifications temporelles pour qu'elles soient considérées comme concordantes). Si l'assertion temporelle emploie **between** et si **entirely** a la valeur **TRUE**, alors l'assertion n'est considérée comme vraie que si la totalité de la période **between** tombe parmi les moments prévus dans la **TimeSpecification** enregistrée.

Exemples

NOTE 3 – Les exemples qui suivent utilisent le format **INTEGER** lorsqu'il existe une possibilité de choix entre **INTEGER** ou **BIT STRING**.

- a) Pour exprimer de 09h00 à 17h00 quotidiennement:

```
periodic {
  timesOfDay { {
    startDayTime hour 9,
    endDayTime hour 17 } } }
```

- b) Pour exprimer tous les lundis:

```
periodic {
  days intDay : {2} }
```

- c) Pour exprimer de 09h00 à 12h00 du lundi au vendredi et toute la journée du samedi en janvier et toute la journée des mardis et vendredis de février et mars:

```
periodic {
  timesOfDay { {
    startDayTime hour 9,
    endDayTime hour 12 } }
  days intDay : {2,3,4,5,6},
  weeks allWeeks : NULL,
  months intMonth : {1} },

  { days {7},
    weeks {1,2,3,4,5},
    months {1} }

  { days {3},
    weeks {1,2,3,4,5},
    months {2,3} } }
```

- d) Pour exprimer août 1996 en totalité:

```
periodic {
  { months {8}
    years {1996} } }
```

- e) Pour exprimer le premier jour de chaque mois:

```
periodic {
  { days {1}
    months NULL } }
```

7.3 Locale Context (contexte de localité)

Le type de contexte *Locale Context* associe une valeur d'attribut à une ou plusieurs localisations telles que les définit POSIX:

```
localeContext CONTEXT ::= {
  WITH SYNTAX LocaleContextSyntax
  ID id-avc-locale }
```

```
LocaleContextSyntax ::= CHOICE {
  localeID1 OBJECT IDENTIFIER,
  localeID2 DirectoryString {ub-localeContextSyntax} }
```

ISO/CEI 9594-6:2001 (F)

On considère qu'il y a accord entre une valeur présentée et une valeur enregistrée si elles sont toutes deux des identificateurs d'objets égaux, ou si elles sont toutes deux des chaînes identiques.

Ne peuvent être utilisés comme valeurs de contexte que des identificateurs d'objets enregistrés ou des chaînes de localisation. Le concept de localisation est défini dans l'ISO/CEI 9945-2:1993, Technologies de l'information – Interface pour la portabilité des systèmes (POSIX) – Partie 2: Enveloppe et services.

NOTE – Il sera créé des autorités d'enregistrement chargées de l'assignation d'OID ou de chaînes d'identification aux spécifications de localité. Le CEN, Comité Européen de Normalisation, a par exemple publié, pour l'enregistrement des informations de localité, la prénorme européenne ENV12005:1996, Procédures pour l'enregistrement des éléments culturels en Europe.

Annexe A

Types d'attributs sélectionnés en ASN.1

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

La présente annexe reprend, sous la forme du module ASN.1 **SelectedAttributeTypes**, toutes les définitions ASN.1 de types et de valeurs contenues dans la présente Spécification d'annuaire.

SelectedAttributeTypes {joint-iso-itu-t ds(5) module(1) selectedAttributeTypes(5) 4}

DEFINITIONS ::=

BEGIN

-- **EXPORTER TOUT** --

-- Les types et les valeurs définis dans le présent module sont exportés en vue de l'utilisation dans les autres modules
 -- ASN.1 contenus dans les Spécifications d'annuaire et en vue de l'utilisation d'autres applications pour accéder aux
 -- services de l'annuaire. D'autres applications peuvent les utiliser pour leur propre compte, mais elles ne doivent pas
 -- gêner les extensions et les modifications nécessaires à la mise à jour et à l'amélioration du service d'annuaire.

IMPORTS

-- de la Rec. UIT-T X.501 | ISO/CEI 9594-2

**directoryAbstractService, id-at, id-avc, id-cat, id-mr, id-not, id-pr, informationFramework,
 serviceAdministration, upperBounds**
FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 4 }

**Attribute, ATTRIBUTE, AttributeType, AttributeValueAssertion, CONTEXT, ContextAssertion,
 DistinguishedName, distinguishedNameMatch, MAPPING-BASED-MATCHING{ },
 MATCHING-RULE, OBJECT-CLASS, objectIdentifierMatch**
FROM InformationFramework informationFramework

AttributeCombination, ContextCombination, MRMapping
FROM ServiceAdministration serviceAdministration

-- de la Rec. UIT-T X.511 | ISO/CEI 9594-3

FilterItem, HierarchySelections, SearchControlOptions, ServiceControlOptions
FROM DirectoryAbstractService directoryAbstractService

-- de la Rec. UIT-T X.520 | ISO/CEI 9594-6

**ub-answerback, ub-business-category, ub-common-name, ub-country-code, ub-description,
 ub-destination-indicator, ub-directory-string-first-component-match, ub-international-isdn-number,
 ub-knowledge-information, ub-localeContextSyntax, ub-locality-name, ub-match, ub-name,
 ub-organization-name, ub-organizational-unit-name, ub-physical-office-name, ub-postal-code,
 ub-postal-line, ub-postal-string, ub-post-office-box, ub-pseudonym, ub-serial-number, ub-state-name,
 ub-street-address, ub-surname, ub-telephone-number, ub-telex-number, ub-teletex-terminal-id,
 ub-title, ub-user-password, ub-x121-address**
FROM UpperBounds upperBounds

-- de la Rec. UIT-T X.411 | ISO/CEI 10021-4

G3FacsimileNonBasicParameters
FROM MTSAbstractService{joint-iso-itu-t mhs(6) mts(3) modules(0)
 mts-abstract-service(1) version-1999(1) } ;

-- Type de chaîne d'annuaire --

DirectoryString { **INTEGER** : maxSize } ::= **CHOICE** {
 teletexString TeletexString (SIZE (1..maxSize)),
 printableString PrintableString (SIZE (1..maxSize)),
 bmpString BMPString (SIZE (1..maxSize)),
 universalString UniversalString (SIZE (1..maxSize)),
 uTF8String UTF8String (SIZE (1..maxSize)) }

-- Type d'attributs --

knowledgeInformation ATTRIBUTE ::= {
WITH SYNTAX DirectoryString {ub-knowledge-information}
EQUALITY MATCHING RULE caseIgnoreMatch
ID id-at-knowledgeInformation }

name ATTRIBUTE ::= {
WITH SYNTAX DirectoryString {ub-name}
EQUALITY MATCHING RULE caseIgnoreMatch
SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
ID id-at-name }

commonName ATTRIBUTE ::= {
SUBTYPE OF name
WITH SYNTAX DirectoryString {ub-common-name}
ID id-at-commonName }

surname ATTRIBUTE ::= {
SUBTYPE OF name
WITH SYNTAX DirectoryString {ub-surname}
ID id-at-surname }

givenName ATTRIBUTE ::= {
SUBTYPE OF name
WITH SYNTAX DirectoryString {ub-name}
ID id-at-givenName }

initials ATTRIBUTE ::= {
SUBTYPE OF name
WITH SYNTAX DirectoryString {ub-name}
ID id-at-initials }

generationQualifier ATTRIBUTE ::= {
SUBTYPE OF name
WITH SYNTAX DirectoryString {ub-name}
ID id-at-generationQualifier }

uniqueIdentifier ATTRIBUTE ::= {
WITH SYNTAX UniqueIdentifier
EQUALITY MATCHING RULE bitStringMatch
ID id-at-uniqueIdentifier }

UniqueIdentifier ::= BIT STRING

dnQualifier ATTRIBUTE ::= {
WITH SYNTAX PrintableString
EQUALITY MATCHING RULE caseIgnoreMatch
ORDERING MATCHING RULE caseIgnoreOrderingMatch
SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
ID id-at-dnQualifier }

serialNumber ATTRIBUTE ::= {
WITH SYNTAX PrintableString (SIZE (1..ub-serial-number))
EQUALITY MATCHING RULE caseIgnoreMatch
SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
ID id-at-serialNumber }

pseudonym ATTRIBUTE ::= {
SUBTYPE OF name
WITH SYNTAX DirectoryString {ub-pseudonym}
ID id-at-pseudonym }

countryName ATTRIBUTE ::= {
 SUBTYPE OF name
 WITH SYNTAX CountryName
 SINGLE VALUE TRUE
 ID id-at-countryName }

CountryName ::= PrintableString (SIZE(2)) -- codes ISO 3166 seulement

localityName ATTRIBUTE ::= {
 SUBTYPE OF name
 WITH SYNTAX DirectoryString {ub-locality-name}
 ID id-at-localityName }

collectiveLocalityName ATTRIBUTE ::= {
 SUBTYPE OF localityName
 COLLECTIVE TRUE
 ID id-at-collectiveLocalityName }

stateOrProvinceName ATTRIBUTE ::= {
 SUBTYPE OF name
 WITH SYNTAX DirectoryString {ub-state-name}
 ID id-at-stateOrProvinceName }

collectiveStateOrProvinceName ATTRIBUTE ::= {
 SUBTYPE OF stateOrProvinceName
 COLLECTIVE TRUE
 ID id-at-collectiveStateOrProvinceName }

streetAddress ATTRIBUTE ::= {
 WITH SYNTAX DirectoryString {ub-street-address}
 EQUALITY MATCHING RULE caseIgnoreMatch
 SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
 ID id-at-streetAddress }

collectiveStreetAddress ATTRIBUTE ::= {
 SUBTYPE OF streetAddress
 COLLECTIVE TRUE
 ID id-at-collectiveStreetAddress }

houseIdentifier ATTRIBUTE ::= {
 WITH SYNTAX DirectoryString {ub-name}
 EQUALITY MATCHING RULE caseIgnoreMatch
 SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
 ID id-at-houseIdentifier }

organizationName ATTRIBUTE ::= {
 SUBTYPE OF name
 WITH SYNTAX DirectoryString {ub-organization-name}
 ID id-at-organizationName }

collectiveOrganizationName ATTRIBUTE ::= {
 SUBTYPE OF organizationName
 COLLECTIVE TRUE
 ID id-at-collectiveOrganizationName }

organizationalUnitName ATTRIBUTE ::= {
 SUBTYPE OF name
 WITH SYNTAX DirectoryString {ub-organizational-unit-name}
 ID id-at-organizationalUnitName }

collectiveOrganizationalUnitName ATTRIBUTE ::= {
 SUBTYPE OF organizationalUnitName
 COLLECTIVE TRUE
 ID id-at-collectiveOrganizationalUnitName }

title ATTRIBUTE ::= {
SUBTYPE OF name
WITH SYNTAX DirectoryString {ub-title}
ID id-at-title }

description ATTRIBUTE ::= {
WITH SYNTAX DirectoryString {ub-description}
EQUALITY MATCHING RULE caseIgnoreMatch
SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
ID id-at-description }

searchGuide ATTRIBUTE ::= {
WITH SYNTAX Guide
ID id-at-searchGuide }

Guide ::= SET {
objectClass [0] OBJECT-CLASS.&id OPTIONAL,
criteria [1] Criteria }

Criteria ::= CHOICE {
type [0] Criterialtem,
and [1] SET OF Criteria,
or [2] SET OF Criteria,
not [3] Criteria }

Criterialtem ::= CHOICE {
equality [0] AttributeType,
substrings [1] AttributeType,
greaterOrEqual [2] AttributeType,
lessOrEqual [3] AttributeType,
approximateMatch [4] AttributeType }

enhancedSearchGuide ATTRIBUTE ::= {
WITH SYNTAX EnhancedGuide
ID id-at-enhancedSearchGuide }

EnhancedGuide ::= SEQUENCE {
objectClass [0] OBJECT-CLASS.&id,
criteria [1] Criteria,
subset [2] INTEGER
{ baseObject (0), oneLevel (1), wholeSubtree (2) } DEFAULT oneLevel }

businessCategory ATTRIBUTE ::= {
WITH SYNTAX DirectoryString {ub-business-category}
EQUALITY MATCHING RULE caseIgnoreMatch
SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
ID id-at-businessCategory }

postalAddress ATTRIBUTE ::= {
WITH SYNTAX PostalAddress
EQUALITY MATCHING RULE caseIgnoreListMatch
SUBSTRINGS MATCHING RULE caseIgnoreListSubstringsMatch
ID id-at-postalAddress }

PostalAddress ::= SEQUENCE SIZE(1..ub-postal-line) OF DirectoryString {ub-postal-string}

collectivePostalAddress ATTRIBUTE ::= {
SUBTYPE OF postalAddress
COLLECTIVE TRUE
ID id-at-collectivePostalAddress }

postalCode ATTRIBUTE ::= {
WITH SYNTAX DirectoryString {ub-postal-code}
EQUALITY MATCHING RULE caseIgnoreMatch
SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
ID id-at-postalCode }

```

collectivePostalCode ATTRIBUTE ::= {
  SUBTYPE OF      postalCode
  COLLECTIVE      TRUE
  ID              id-at-collectivePostalCode }

postOfficeBox ATTRIBUTE ::= {
  WITH SYNTAX      DirectoryString {ub-post-office-box}
  EQUALITY MATCHING RULE  caseIgnoreMatch
  SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
  ID              id-at-postOfficeBox }

collectivePostOfficeBox ATTRIBUTE ::= {
  SUBTYPE OF      postOfficeBox
  COLLECTIVE      TRUE
  ID              id-at-collectivePostOfficeBox }

physicalDeliveryOfficeName ATTRIBUTE ::= {
  WITH SYNTAX      DirectoryString {ub-physical-office-name}
  EQUALITY MATCHING RULE  caseIgnoreMatch
  SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
  ID              id-at-physicalDeliveryOfficeName }

collectivePhysicalDeliveryOfficeName ATTRIBUTE ::= {
  SUBTYPE OF      physicalDeliveryOfficeName
  COLLECTIVE      TRUE
  ID              id-at-collectivePhysicalDeliveryOfficeName }

telephoneNumber ATTRIBUTE ::= {
  WITH SYNTAX      TelephoneNumber
  EQUALITY MATCHING RULE  telephoneNumberMatch
  SUBSTRINGS MATCHING RULE  telephoneNumberSubstringsMatch
  ID              id-at-telephoneNumber }

TelephoneNumber ::= PrintableString (SIZE(1..ub-telephone-number))
  -- Chaîne conforme à la Rec. UIT-T E.123 seulement

collectiveTelephoneNumber ATTRIBUTE ::= {
  SUBTYPE OF      telephoneNumber
  COLLECTIVE      TRUE
  ID              id-at-collectiveTelephoneNumber }

telexNumber ATTRIBUTE ::= {
  WITH SYNTAX      TelexNumber
  ID              id-at-telexNumber }

TelexNumber ::= SEQUENCE {
  telexNumber      PrintableString (SIZE (1..ub-telex-number)),
  countryCode      PrintableString (SIZE (1..ub-country-code)),
  answerback      PrintableString (SIZE (1..ub-answerback)) }

collectiveTelexNumber ATTRIBUTE ::= {
  SUBTYPE OF      telexNumber
  COLLECTIVE      TRUE
  ID              id-at-collectiveTelexNumber }

facsimileTelephoneNumber ATTRIBUTE ::= {
  WITH SYNTAX      FacsimileTelephoneNumber
  ID              id-at-facsimileTelephoneNumber }

FacsimileTelephoneNumber ::= SEQUENCE {
  telephoneNumberTelephoneNumber,
  parameters      G3FacsimileNonBasicParameters OPTIONAL }

collectiveFacsimileTelephoneNumber ATTRIBUTE ::= {
  SUBTYPE OF      facsimileTelephoneNumber
  COLLECTIVE      TRUE
  ID              id-at-collectiveFacsimileTelephoneNumber }

```

x121Address ATTRIBUTE ::= {
WITH SYNTAX X121Address
EQUALITY MATCHING RULE numericStringMatch
SUBSTRINGS MATCHING RULE numericStringSubstringsMatch
ID id-at-x121Address }

X121Address ::= NumericString (SIZE(1..ub-x121-address))
-- Chaîne conforme à la Rec. UIT-T X.121

internationalISDNNumber ATTRIBUTE ::= {
WITH SYNTAX InternationalISDNNumber
EQUALITY MATCHING RULE numericStringMatch
SUBSTRINGS MATCHING RULE numericStringSubstringsMatch
ID id-at-internationalISDNNumber }

InternationalISDNNumber ::= NumericString (SIZE(1..ub-international-isdn-number))
-- Chaîne conforme à la Rec. UIT-T E.164 seulement

collectiveInternationalISDNNumber ATTRIBUTE ::= {
SUBTYPE OF internationalISDNNumber
COLLECTIVE TRUE
ID id-at-collectiveInternationalISDNNumber }

registeredAddress ATTRIBUTE ::= {
SUBTYPE OF postalAddress
WITH SYNTAX PostalAddress
ID id-at-registeredAddress }

destinationIndicator ATTRIBUTE ::= {
WITH SYNTAX DestinationIndicator
EQUALITY MATCHING RULE caseIgnoreMatch
SUBSTRINGS MATCHING RULE caseIgnoreSubstringsMatch
ID id-at-destinationIndicator }

DestinationIndicator ::= PrintableString (SIZE(1..ub-destination-indicator))
-- caractères alphabétiques seulement

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

communicationsNetwork ATTRIBUTE ::= {
WITH SYNTAX OBJECT IDENTIFIER
EQUALITY MATCHING RULE objectIdentifierMatch
SINGLE VALUE TRUE
ID id-at-communicationsNetwork }

preferredDeliveryMethod ATTRIBUTE ::= {
WITH SYNTAX PreferredDeliveryMethod
SINGLE VALUE TRUE
ID id-at-preferredDeliveryMethod }

PreferredDeliveryMethod ::= SEQUENCE OF INTEGER {
any-delivery-method (0),
mhs-delivery (1),
physical-delivery (2),
telex-delivery (3),
teletex-delivery (4),
g3-facsimile-delivery (5),
g4-facsimile-delivery (6),
ia5-terminal-delivery (7),
videotex-delivery (8),
telephone-delivery (9) }


```

presentationAddress ATTRIBUTE ::= {
    WITH SYNTAX                PresentationAddress
    EQUALITY MATCHING RULE     presentationAddressMatch
    SINGLE VALUE               TRUE
    ID                          id-at-presentationAddress }

PresentationAddress ::= SEQUENCE {
    pSelector    [0]  OCTET STRING OPTIONAL,
    sSelector    [1]  OCTET STRING OPTIONAL,
    tSelector    [2]  OCTET STRING OPTIONAL,
    nAddresses   [3]  SET SIZE (1..MAX) OF OCTET STRING }

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

protocollInformation ATTRIBUTE ::= {
    WITH SYNTAX                ProtocollInformation
    EQUALITY MATCHING RULE     protocollInformationMatch
    ID                          id-at-protocollInformation }

ProtocollInformation ::= SEQUENCE {
    nAddress      OCTET STRING,
    profiles      SET OF OBJECT IDENTIFIER }

distinguishedName ATTRIBUTE ::= {
    WITH SYNTAX                DistinguishedName
    EQUALITY MATCHING RULE     distinguishedNameMatch
    ID                          id-at-distinguishedName }

member ATTRIBUTE ::= {
    SUBTYPE OF                 distinguishedName
    ID                          id-at-member }

uniqueMember ATTRIBUTE ::= {
    WITH SYNTAX                NameAndOptionalUID
    EQUALITY MATCHING RULE     uniqueMemberMatch
    ID                          id-at-uniqueMember }

NameAndOptionalUID ::= SEQUENCE {
    dn              DistinguishedName,
    uid             UniqueIdentifier OPTIONAL }

owner ATTRIBUTE ::= {
    SUBTYPE OF                 distinguishedName
    ID                          id-at-owner }

roleOccupant ATTRIBUTE ::= {
    SUBTYPE OF                 distinguishedName
    ID                          id-at-roleOccupant }

seeAlso ATTRIBUTE ::= {
    SUBTYPE OF                 distinguishedName
    ID                          id-at-seeAlso }

dmdName ATTRIBUTE ::= {
    SUBTYPE OF                 name
    WITH SYNTAX                DirectoryString{ub-common-name}
    ID                          id-at-dmdName }

-- Attributs de notification --

dSAPProblem ATTRIBUTE ::= {
    WITH SYNTAX                OBJECT IDENTIFIER
    EQUALITY MATCHING RULE     objectIdentifierMatch
    ID                          id-not-dSAPProblem }

```

```

searchServiceProblem ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    SINGLE VALUE        TRUE
    ID                  id-not-searchServiceProblem }

serviceType ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    SINGLE VALUE        TRUE
    ID                  id-not-serviceType }

attributeTypeList ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                  id-not-attributeTypeList }

matchingRuleList ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                  id-not-matchingRuleList }

filterItem ATTRIBUTE ::= {
    WITH SYNTAX          FilterItem
    ID                  id-not-filterItem }

attributeCombinations ATTRIBUTE ::= {
    WITH SYNTAX          AttributeCombination
    ID                  id-not-attributeCombinations }

contextTypeList ATTRIBUTE ::= {
    WITH SYNTAX          OBJECT IDENTIFIER
    EQUALITY MATCHING RULE objectIdentifierMatch
    ID                  id-not-contextTypeList }

contextList ATTRIBUTE ::= {
    WITH SYNTAX          ContextAssertion
    ID                  id-not-contextList }

contextCombinations ATTRIBUTE ::= {
    WITH SYNTAX          ContextCombination
    ID                  id-not-contextCombinations }

hierarchySelectList ATTRIBUTE ::= {
    WITH SYNTAX          HierarchySelections
    SINGLE VALUE        TRUE
    ID                  id-not-hierarchySelectList }

searchControlOptionsList ATTRIBUTE ::= {
    WITH SYNTAX          SearchControlOptions
    SINGLE VALUE        TRUE
    ID                  id-not-searchControlOptionsList }

serviceControlOptionsList ATTRIBUTE ::= {
    WITH SYNTAX          ServiceControlOptions
    SINGLE VALUE        TRUE
    ID                  id-not-serviceControlOptionsList }

multipleMatchingLocalities ATTRIBUTE ::= {
    WITH SYNTAX          MultipleMatchingLocalities
    ID                  id-not-multipleMatchingLocalities }

MultipleMatchingLocalities ::= SEQUENCE {
    matchingRuleUsed     MATCHING-RULE.&id OPTIONAL,
    attributeList        SEQUENCE OF AttributeValueAssertion }

```

proposedRelaxation ATTRIBUTE ::= {
 WITH SYNTAX SEQUENCE OF MRMapping
 ID id-not-proposedRelaxation }

appliedRelaxation ATTRIBUTE ::= {
 WITH SYNTAX OBJECT IDENTIFIER
 EQUALITY MATCHING RULE objectIdentifierMatch
 ID id-not-appliedRelaxation }

-- Règles de concordance --

caseIgnoreMatch MATCHING-RULE ::= {
 SYNTAX DirectoryString {ub-match}
 ID id-mr-caseIgnoreMatch }

caseIgnoreOrderingMatch MATCHING-RULE ::= {
 SYNTAX DirectoryString {ub-match}
 ID id-mr-caseIgnoreOrderingMatch }

caseIgnoreSubstringsMatch MATCHING-RULE ::= {
 SYNTAX SubstringAssertion
 ID id-mr-caseIgnoreSubstringsMatch }

SubstringAssertion ::= SEQUENCE OF CHOICE {
 initial [0] DirectoryString {ub-match},
 any [1] DirectoryString {ub-match},
 final [2] DirectoryString {ub-match},
 control Attribute } -- Elément utilisé pour spécifier l'interprétation des éléments suivants
 -- au plus un composant initial et un composant final

caseExactMatch MATCHING-RULE ::= {
 SYNTAX DirectoryString {ub-match}
 ID id-mr-caseExactMatch }

caseExactOrderingMatch MATCHING-RULE ::= {
 SYNTAX DirectoryString {ub-match}
 ID id-mr-caseExactOrderingMatch }

caseExactSubstringsMatch MATCHING-RULE ::= {
 SYNTAX SubstringAssertion -- choix PrintableString seulement
 ID id-mr-caseExactSubstringsMatch }

numericStringMatch MATCHING-RULE ::= {
 SYNTAX NumericString
 ID id-mr-numericStringMatch }

numericStringOrderingMatch MATCHING-RULE ::= {
 SYNTAX NumericString
 ID id-mr-numericStringOrderingMatch }

numericStringSubstringsMatch MATCHING-RULE ::= {
 SYNTAX SubstringAssertion
 ID id-mr-numericStringSubstringsMatch }

caseIgnoreListMatch MATCHING-RULE ::= {
 SYNTAX CaseIgnoreList
 ID id-mr-caseIgnoreListMatch }

CaseIgnoreList ::= SEQUENCE OF DirectoryString {ub-match}

caseIgnoreListSubstringsMatch MATCHING-RULE ::= {
 SYNTAX SubstringAssertion
 ID id-mr-caseIgnoreListSubstringsMatch }

storedPrefixMatch MATCHING-RULE ::= {
 SYNTAX DirectoryString {ub-match}
 ID id-mr-storedPrefixMatch }

```

booleanMatch MATCHING-RULE ::= {
  SYNTAX   BOOLEAN
  ID       id-mr-booleanMatch }

integerMatch MATCHING-RULE ::= {
  SYNTAX   INTEGER
  ID       id-mr-integerMatch }

integerOrderingMatch MATCHING-RULE ::= {
  SYNTAX   INTEGER
  ID       id-mr-integerOrderingMatch }

bitStringMatch MATCHING-RULE ::= {
  SYNTAX   BIT STRING
  ID       id-mr-bitStringMatch }

octetStringMatch MATCHING-RULE ::= {
  SYNTAX   OCTET STRING
  ID       id-mr-octetStringMatch }

octetStringOrderingMatch MATCHING-RULE ::= {
  SYNTAX   OCTET STRING
  ID       id-mr-octetStringOrderingMatch }

octetStringSubstringsMatch MATCHING-RULE ::= {
  SYNTAX   OctetSubstringAssertion
  ID       id-mr-octetStringSubstringsMatch }

OctetSubstringAssertion ::= SEQUENCE OF CHOICE {
  initial   [0]  OCTET STRING,
  any       [1]  OCTET STRING,
  final     [2]  OCTET STRING }
-- au plus un composant initial et un composant final

telephoneNumberMatch MATCHING-RULE ::= {
  SYNTAX   PrintableString
  ID       id-mr-telephoneNumberMatch }

telephoneNumberSubstringsMatch MATCHING-RULE ::= {
  SYNTAX   SubstringAssertion
  ID       id-mr-telephoneNumberSubstringsMatch }

presentationAddressMatch MATCHING-RULE ::= {
  SYNTAX   PresentationAddress
  ID       id-mr-presentationAddressMatch }

uniqueMemberMatch MATCHING-RULE ::= {
  SYNTAX   NameAndOptionalUID
  ID       id-mr-uniqueMemberMatch }

protocolInformationMatch MATCHING-RULE ::= {
  SYNTAX   OCTET STRING
  ID       id-mr-protocolInformationMatch }

uTCTimeMatch MATCHING-RULE ::= {
  SYNTAX   UTCTime
  ID       id-mr-uTCTimeMatch }

uTCTimeOrderingMatch MATCHING-RULE ::= {
  SYNTAX   UTCTime
  ID       id-mr-uTCTimeOrderingMatch }

generalizedTimeMatch MATCHING-RULE ::= {
  SYNTAX   GeneralizedTime
  ID       id-mr-generalizedTimeMatch }
-- selon le § 41.3 b) ou c) de la Rec. UIT-T X.680 | ISO/CEI 8824-1

```

```

generalizedTimeOrderingMatch MATCHING-RULE ::= {
  SYNTAX   GeneralizedTime
           -- selon le § 41.3 b) ou c) de la Rec. UIT-T X.680 | ISO/CEI 8824-1
  ID       id-mr-generalizedTimeOrderingMatch }

systemProposedMatch MATCHING-RULE ::= {
  ID       id-mr-systemProposedMatch }

integerFirstComponentMatch MATCHING-RULE ::= {
  SYNTAX   INTEGER
  ID       id-mr-integerFirstComponentMatch }

objectIdentifierFirstComponentMatch MATCHING-RULE ::= {
  SYNTAX   OBJECT IDENTIFIER
  ID       id-mr-objectIdentifierFirstComponentMatch }

directoryStringFirstComponentMatch MATCHING-RULE ::= {
  SYNTAX   DirectoryString {ub-directory-string-first-component-match}
  ID       id-mr-directoryStringFirstComponentMatch }

wordMatch MATCHING-RULE ::= {
  SYNTAX   DirectoryString {ub-match}
  ID       id-mr-wordMatch }

keywordMatch MATCHING-RULE ::= {
  SYNTAX   DirectoryString {ub-match}
  ID       id-mr-keywordMatch }

generalWordMatch MATCHING-RULE ::= {
  SYNTAX   SubstringAssertion
  ID       id-mr-generalWordMatch }

sequenceMatchType ATTRIBUTE ::= {
  WITH SYNTAX   SequenceMatchType
  SINGLE VALUE TRUE
  ID           id-cat-sequenceMatchType } -- valeur par défaut sequenceExact

SequenceMatchType ::= ENUMERATED {
  sequenceExact           (0),
  sequenceDeletion        (1),
  sequenceRestrictedDeletion (2),
  sequencePermutation      (3),
  sequencePermutationAndDeletion (4),
  sequenceProviderDefined  (5) }

wordMatchTypes ATTRIBUTE ::= {
  WITH SYNTAX   WordMatchTypes
  SINGLE VALUE TRUE
  ID           id-cat-wordMatchType } -- valeur par défaut wordExact

WordMatchTypes ::= ENUMERATED {
  wordExact           (0),
  wordTruncated       (1),
  wordPhonetic        (2),
  wordProviderDefined (3) }

characterMatchTypes ATTRIBUTE ::= {
  WITH SYNTAX   CharacterMatchTypes
  SINGLE VALUE TRUE
  ID           id-cat-characterMatchTypes }

CharacterMatchTypes ::= ENUMERATED {
  characterExact           (0),
  characterCaseIgnore      (1),
  characterMapped          (2) }

```

```
selectedContexts ATTRIBUTE ::= {
    WITH SYNTAX    ContextAssertion
    ID             id-cat-selectedContexts }
```

```
approximateStringMatch MATCHING-RULE ::= {
    ID             id-mr-approximateStringMatch }
```

```
ignoreIfAbsentMatch MATCHING-RULE ::= {
    ID             id-mr-ignoreIfAbsentMatch }
```

```
nullMatch MATCHING-RULE ::= {
    ID             id-mr-nullMatch }
```

ZONAL-MATCHING ::= MAPPING-BASED-MATCHING { ZonalSelect, TRUE, ZonalResult, zonalMatch }

ZonalSelect ::= SEQUENCE OF AttributeType

```
ZonalResult ::= ENUMERATED {
    cannot-select-mapping    (0),
    zero-mappings           (2),
    multiple-mappings       (3) }
```

```
zonalMatch MATCHING-RULE ::= {
    UNIQUE-MATCH-INDICATOR    multipleMatchingLocalities
    ID                         id-mr-zonalMatch }
```

-- Contextes --

```
languageContext CONTEXT ::= {
    WITH SYNTAX    LanguageContextSyntax
    ID             id-avc-language }
```

LanguageContextSyntax ::= PrintableString (SIZE(2..3)) -- codes ISO 639-2 seulement

```
temporalContext CONTEXT ::= {
    WITH SYNTAX    TimeSpecification
    ASSERTED AS    TimeAssertion
    ID             id-avc-temporal }
```

```
TimeSpecification ::= SEQUENCE {
    time             CHOICE {
        absolute     SEQUENCE {
            startTime [0] GeneralizedTime OPTIONAL,
            endTime   [1] GeneralizedTime OPTIONAL },
        periodic     SET OF Period },
    notThisTime     BOOLEAN DEFAULT FALSE,
    timeZone        TimeZone OPTIONAL }
```

```
Period ::= SEQUENCE {
    timesOfDay [0] SET SIZE (1..MAX) OF DayTimeBand OPTIONAL,
    days       [1] CHOICE {
        intDay  SET OF INTEGER,
        bitDay  BIT STRING { sunday (0), monday (1), tuesday (2),
            wednesday (3), thursday (4), friday (5), saturday (6) },
        dayOf   XDayOf } OPTIONAL,
    weeks      [2] CHOICE {
        allWeeks  NULL,
        intWeek   SET OF INTEGER,
        bitWeek   BIT STRING { week1 (0), week2 (1), week3 (2), week4 (3),
            week5 (4) } } OPTIONAL,
    months     [3] CHOICE {
        allMonths  NULL,
        intMonth   SET OF INTEGER,
        bitMonth   BIT STRING { january (0), february (1), march (2), april (3),
            may (4), june (5), july (6), august (7), september (8),
            october (9), november (10), december (11) }
    } OPTIONAL,
```

years [4] SET OF INTEGER (1000 .. MAX) OPTIONAL }

XDayOf ::= CHOICE {
 first [1] NamedDay,
 second [2] NamedDay,
 third [3] NamedDay,
 fourth [4] NamedDay,
 fifth [5] NamedDay }

NamedDay ::= CHOICE {
 intNamedDays ENUMERATED {
 sunday (1),
 monday (2),
 tuesday (3),
 wednesday (4),
 thursday (5),
 friday (6),
 saturday (7) },
 bitNamedDays BIT STRING { sunday (0), monday (1) , tuesday (2),
 wednesday (3), thursday (4), friday (5), saturday (6) } }

DayTimeBand ::= SEQUENCE {
 startDayTime [0] DayTime DEFAULT { hour 0 },
 endDayTime [1] DayTime DEFAULT { hour 23, minute 59, second 59 } }

DayTime ::= SEQUENCE {
 hour [0] INTEGER (0..23),
 minute [1] INTEGER (0..59) DEFAULT 0,
 second [2] INTEGER (0..59) DEFAULT 0 }

TimeZone ::= INTEGER (-12..12)

TimeAssertion ::= CHOICE {
 now NULL,
 at GeneralizedTime,
 between SEQUENCE {
 startTime [0] GeneralizedTime,
 endTime [1] GeneralizedTime OPTIONAL,
 entirely BOOLEAN DEFAULT FALSE } }

localeContext CONTEXT ::= {
 WITH SYNTAX LocaleContextSyntax
 ID id-avc-locale }

LocaleContextSyntax ::= CHOICE {
 localeID1 OBJECT IDENTIFIER,
 localeID2 DirectoryString {ub-localeContextSyntax} }

-- Affectation d'identificateurs d'objets --

-- les identificateurs d'objets affectés dans d'autres modules sont indiqués dans les commentaires

-- Attributs --

-- id-at-objectClass	OBJECT IDENTIFIER ::= {id-at 0}
-- id-at-aliasedEntryName	OBJECT IDENTIFIER ::= {id-at 1}
id-at-encryptedAliasedEntryName	OBJECT IDENTIFIER ::= {id-at 1 2}
id-at-knowledgeInformation	OBJECT IDENTIFIER ::= {id-at 2}
id-at-commonName	OBJECT IDENTIFIER ::= {id-at 3}
id-at-encryptedCommonName	OBJECT IDENTIFIER ::= {id-at 3 2}
id-at-surname	OBJECT IDENTIFIER ::= {id-at 4}
id-at-encryptedSurname	OBJECT IDENTIFIER ::= {id-at 4 2}
id-at-serialNumber	OBJECT IDENTIFIER ::= {id-at 5}
id-at-encryptedSerialNumber	OBJECT IDENTIFIER ::= {id-at 5 2}
id-at-countryName	OBJECT IDENTIFIER ::= {id-at 6}
id-at-encryptedCountryName	OBJECT IDENTIFIER ::= {id-at 6 2}
id-at-localityName	OBJECT IDENTIFIER ::= {id-at 7}
id-at-encryptedLocalityName	OBJECT IDENTIFIER ::= {id-at 7 2}

id-at-collectiveLocalityName	OBJECT IDENTIFIER	::=	{id-at 7 1}
id-at-encryptedCollectiveLocalityName	OBJECT IDENTIFIER	::=	{id-at 7 1 2}
id-at-stateOrProvinceName	OBJECT IDENTIFIER	::=	{id-at 8}
id-at-encryptedStateOrProvinceName	OBJECT IDENTIFIER	::=	{id-at 8 2}
id-at-collectiveStateOrProvinceName	OBJECT IDENTIFIER	::=	{id-at 8 1}
id-at-encryptedCollectiveStateOrProvinceName	OBJECT IDENTIFIER	::=	{id-at 8 1 2}
id-at-streetAddress	OBJECT IDENTIFIER	::=	{id-at 9}
id-at-encryptedStreetAddress	OBJECT IDENTIFIER	::=	{id-at 9 2}
id-at-collectiveStreetAddress	OBJECT IDENTIFIER	::=	{id-at 9 1}
id-at-encryptedCollectiveStreetAddress	OBJECT IDENTIFIER	::=	{id-at 9 1 2}
id-at-organizationName	OBJECT IDENTIFIER	::=	{id-at 10}
id-at-encryptedOrganizationName	OBJECT IDENTIFIER	::=	{id-at 10 2}
id-at-collectiveOrganizationName	OBJECT IDENTIFIER	::=	{id-at 10 1}
id-at-encryptedCollectiveOrganizationName	OBJECT IDENTIFIER	::=	{id-at 10 1 2}
id-at-organizationalUnitName	OBJECT IDENTIFIER	::=	{id-at 11}
id-at-encryptedOrganizationalUnitName	OBJECT IDENTIFIER	::=	{id-at 11 2}
id-at-collectiveOrganizationalUnitName	OBJECT IDENTIFIER	::=	{id-at 11 1}
id-at-encryptedCollectiveOrganizationalUnitName	OBJECT IDENTIFIER	::=	{id-at 11 1 2}
id-at-title	OBJECT IDENTIFIER	::=	{id-at 12}
id-at-encryptedTitle	OBJECT IDENTIFIER	::=	{id-at 12 2}
id-at-description	OBJECT IDENTIFIER	::=	{id-at 13}
id-at-encryptedDescription	OBJECT IDENTIFIER	::=	{id-at 13 2}
id-at-searchGuide	OBJECT IDENTIFIER	::=	{id-at 14}
id-at-encryptedSearchGuide	OBJECT IDENTIFIER	::=	{id-at 14 2}
id-at-businessCategory	OBJECT IDENTIFIER	::=	{id-at 15}
id-at-encryptedBusinessCategory	OBJECT IDENTIFIER	::=	{id-at 15 2}
id-at-postalAddress	OBJECT IDENTIFIER	::=	{id-at 16}
id-at-encryptedPostalAddress	OBJECT IDENTIFIER	::=	{id-at 16 2}
id-at-collectivePostalAddress	OBJECT IDENTIFIER	::=	{id-at 16 1}
id-at-encryptedCollectivePostalAddress	OBJECT IDENTIFIER	::=	{id-at 16 1 2}
id-at-postalCode	OBJECT IDENTIFIER	::=	{id-at 17}
id-at-encryptedPostalCode	OBJECT IDENTIFIER	::=	{id-at 17 2}
id-at-collectivePostalCode	OBJECT IDENTIFIER	::=	{id-at 17 1}
id-at-encryptedCollectivePostalCode	OBJECT IDENTIFIER	::=	{id-at 17 1 2}
id-at-postOfficeBox	OBJECT IDENTIFIER	::=	{id-at 18}
id-at-encryptedPostOfficeBox	OBJECT IDENTIFIER	::=	{id-at 18 2}
id-at-collectivePostOfficeBox	OBJECT IDENTIFIER	::=	{id-at 18 1}
id-at-encryptedCollectivePostOfficeBox	OBJECT IDENTIFIER	::=	{id-at 18 1 2}
id-at-physicalDeliveryOfficeName	OBJECT IDENTIFIER	::=	{id-at 19}
id-at-encryptedPhysicalDeliveryOfficeName	OBJECT IDENTIFIER	::=	{id-at 19 2}
id-at-collectivePhysicalDeliveryOfficeName	OBJECT IDENTIFIER	::=	{id-at 19 1}
id-at-encryptedCollectivePhysicalDeliveryOfficeName	OBJECT IDENTIFIER	::=	{id-at 19 1 2}
id-at-telephoneNumber	OBJECT IDENTIFIER	::=	{id-at 20}
id-at-encryptedTelephoneNumber	OBJECT IDENTIFIER	::=	{id-at 20 2}
id-at-collectiveTelephoneNumber	OBJECT IDENTIFIER	::=	{id-at 20 1}
id-at-encryptedCollectiveTelephoneNumber	OBJECT IDENTIFIER	::=	{id-at 20 1 2}
id-at-telexNumber	OBJECT IDENTIFIER	::=	{id-at 21}
id-at-encryptedTelexNumber	OBJECT IDENTIFIER	::=	{id-at 21 2}
id-at-collectiveTelexNumber	OBJECT IDENTIFIER	::=	{id-at 21 1}
id-at-encryptedCollectiveTelexNumber	OBJECT IDENTIFIER	::=	{id-at 21 1 2}
-- id-at-teletexTerminalIdentifier	OBJECT IDENTIFIER	::=	{id-at 22}
-- id-at-encryptedTeletexTerminalIdentifier	OBJECT IDENTIFIER	::=	{id-at 22 2}
-- id-at-collectiveTeletexTerminalIdentifier	OBJECT IDENTIFIER	::=	{id-at 22 1}
-- id-at-encryptedCollectiveTeletexTerminalIdentifier	OBJECT IDENTIFIER	::=	{id-at 22 1 2}
id-at-facsimileTelephoneNumber	OBJECT IDENTIFIER	::=	{id-at 23}
id-at-encryptedFacsimileTelephoneNumber	OBJECT IDENTIFIER	::=	{id-at 23 2}
id-at-collectiveFacsimileTelephoneNumber	OBJECT IDENTIFIER	::=	{id-at 23 1}
id-at-encryptedCollectiveFacsimileTelephoneNumber	OBJECT IDENTIFIER	::=	{id-at 23 1 2}
id-at-x121Address	OBJECT IDENTIFIER	::=	{id-at 24}
id-at-encryptedX121Address	OBJECT IDENTIFIER	::=	{id-at 24 2}
id-at-internationalISDNNumber	OBJECT IDENTIFIER	::=	{id-at 25}
id-at-encryptedInternationalISDNNumber	OBJECT IDENTIFIER	::=	{id-at 25 2}
id-at-collectiveInternationalISDNNumber	OBJECT IDENTIFIER	::=	{id-at 25 1}

id-at-encryptedCollectiveInternationalISDNNumber	OBJECT IDENTIFIER	::=	{id-at 25 1 2}
id-at-registeredAddress	OBJECT IDENTIFIER	::=	{id-at 26}
id-at-encryptedRegisteredAddress	OBJECT IDENTIFIER	::=	{id-at 26 2}
id-at-destinationIndicator	OBJECT IDENTIFIER	::=	{id-at 27}
id-at-encryptedDestinationIndicator	OBJECT IDENTIFIER	::=	{id-at 27 2}
id-at-preferredDeliveryMethod	OBJECT IDENTIFIER	::=	{id-at 28}
id-at-encryptedPreferredDeliveryMethod	OBJECT IDENTIFIER	::=	{id-at 28 2}
id-at-presentationAddress	OBJECT IDENTIFIER	::=	{id-at 29}
id-at-encryptedPresentationAddress	OBJECT IDENTIFIER	::=	{id-at 29 2}
id-at-supportedApplicationContext	OBJECT IDENTIFIER	::=	{id-at 30}
id-at-encryptedSupportedApplicationContext	OBJECT IDENTIFIER	::=	{id-at 30 2}
id-at-member	OBJECT IDENTIFIER	::=	{id-at 31}
id-at-encryptedMember	OBJECT IDENTIFIER	::=	{id-at 31 2}
id-at-owner	OBJECT IDENTIFIER	::=	{id-at 32}
id-at-encryptedOwner	OBJECT IDENTIFIER	::=	{id-at 32 2}
id-at-roleOccupant	OBJECT IDENTIFIER	::=	{id-at 33}
id-at-encryptedRoleOccupant	OBJECT IDENTIFIER	::=	{id-at 33 2}
id-at-seeAlso	OBJECT IDENTIFIER	::=	{id-at 34}
id-at-encryptedSeeAlso	OBJECT IDENTIFIER	::=	{id-at 34 2}
<i>-- id-at-userPassword</i>	OBJECT IDENTIFIER	::=	{id-at 35}
id-at-encryptedUserPassword	OBJECT IDENTIFIER	::=	{id-at 35 2}
<i>-- id-at-userCertificate</i>	OBJECT IDENTIFIER	::=	{id-at 36}
id-at-encryptedUserCertificate	OBJECT IDENTIFIER	::=	{id-at 36 2}
<i>-- id-at-cACertificate</i>	OBJECT IDENTIFIER	::=	{id-at 37}
id-at-encryptedCACertificate	OBJECT IDENTIFIER	::=	{id-at 37 2}
<i>-- id-at-authorityRevocationList</i>	OBJECT IDENTIFIER	::=	{id-at 38}
id-at-encryptedAuthorityRevocationList	OBJECT IDENTIFIER	::=	{id-at 38 2}
<i>-- id-at-certificateRevocationList</i>	OBJECT IDENTIFIER	::=	{id-at 39}
id-at-encryptedCertificateRevocationList	OBJECT IDENTIFIER	::=	{id-at 39 2}
<i>-- id-at-crossCertificatePair</i>	OBJECT IDENTIFIER	::=	{id-at 40}
id-at-encryptedCrossCertificatePair	OBJECT IDENTIFIER	::=	{id-at 40 2}
id-at-name	OBJECT IDENTIFIER	::=	{id-at 41}
id-at-givenName	OBJECT IDENTIFIER	::=	{id-at 42}
id-at-encryptedGivenName	OBJECT IDENTIFIER	::=	{id-at 42 2}
id-at-initials	OBJECT IDENTIFIER	::=	{id-at 43}
id-at-encryptedInitials	OBJECT IDENTIFIER	::=	{id-at 43 2}
id-at-generationQualifier	OBJECT IDENTIFIER	::=	{id-at 44}
id-at-encryptedGenerationQualifier	OBJECT IDENTIFIER	::=	{id-at 44 2}
id-at-uniqueIdentifier	OBJECT IDENTIFIER	::=	{id-at 45}
id-at-encryptedUniqueIdentifier	OBJECT IDENTIFIER	::=	{id-at 45 2}
id-at-dnQualifier	OBJECT IDENTIFIER	::=	{id-at 46}
id-at-encryptedDnQualifier	OBJECT IDENTIFIER	::=	{id-at 46 2}
id-at-enhancedSearchGuide	OBJECT IDENTIFIER	::=	{id-at 47}
id-at-encryptedEnhancedSearchGuide	OBJECT IDENTIFIER	::=	{id-at 47 2}
id-at-protocolInformation	OBJECT IDENTIFIER	::=	{id-at 48}
id-at-encryptedProtocolInformation	OBJECT IDENTIFIER	::=	{id-at 48 2}
id-at-distinguishedName	OBJECT IDENTIFIER	::=	{id-at 49}
id-at-encryptedDistinguishedName	OBJECT IDENTIFIER	::=	{id-at 49 2}
id-at-uniqueMember	OBJECT IDENTIFIER	::=	{id-at 50}
id-at-encryptedUniqueMember	OBJECT IDENTIFIER	::=	{id-at 50 2}
id-at-houseIdentifier	OBJECT IDENTIFIER	::=	{id-at 51}
id-at-encryptedHouseIdentifier	OBJECT IDENTIFIER	::=	{id-at 51 2}
<i>-- id-at-supportedAlgorithms</i>	OBJECT IDENTIFIER	::=	{id-at 52}
id-at-encryptedSupportedAlgorithms	OBJECT IDENTIFIER	::=	{id-at 52 2}
<i>-- id-at-deltaRevocationList</i>	OBJECT IDENTIFIER	::=	{id-at 53}
id-at-encryptedDeltaRevocationList	OBJECT IDENTIFIER	::=	{id-at 53 2}
id-at-dmdName	OBJECT IDENTIFIER	::=	{id-at 54}
id-at-encryptedDmdName	OBJECT IDENTIFIER	::=	{id-at 54 2}
<i>-- id-at-clearance</i>	OBJECT IDENTIFIER	::=	{id-at 55}
id-at-encryptedClearance	OBJECT IDENTIFIER	::=	{id-at 55 2}
<i>-- id-at-defaultDirQop</i>	OBJECT IDENTIFIER	::=	{id-at 56}
<i>-- id-at-encryptedDefaultDirQop</i>	OBJECT IDENTIFIER	::=	{id-at 56 2}
<i>-- id-at-attributeIntegrityInfo</i>	OBJECT IDENTIFIER	::=	{id-at 57}
id-at-encryptedAttributeIntegrityInfo	OBJECT IDENTIFIER	::=	{id-at 57 2}
<i>-- id-at-attributeCertificate</i>	OBJECT IDENTIFIER	::=	{id-at 58}
id-at-encryptedAttributeCertificate	OBJECT IDENTIFIER	::=	{id-at 58 2}
<i>-- id-at-attributeCertificateRevocationList</i>	OBJECT IDENTIFIER	::=	{id-at 59}

id-at-encryptedAttributeCertificateRevocationList

<i>-- id-at-confKeyInfo</i>	OBJECT IDENTIFIER ::= {id-at 59 2}
id-at-encryptedConfKeyInfo	OBJECT IDENTIFIER ::= {id-at 60}
<i>-- id-at-aACertificate</i>	OBJECT IDENTIFIER ::= {id-at 60 2}
<i>-- id-at-attributeDescriptorCertificate</i>	OBJECT IDENTIFIER ::= {id-at 61}
<i>-- id-at-attributeAuthorityRevocationList</i>	OBJECT IDENTIFIER ::= {id-at 62}
<i>-- id-at-family-information</i>	OBJECT IDENTIFIER ::= {id-at 63}
id-at-pseudonym	OBJECT IDENTIFIER ::= {id-at 64}
id-at-communicationsService	OBJECT IDENTIFIER ::= {id-at 65}
id-at-communicationsNetwork	OBJECT IDENTIFIER ::= {id-at 66}
<i>-- id-at-certificationPracticeStmnt</i>	OBJECT IDENTIFIER ::= {id-at 67}
<i>-- id-at-certificatePolicy</i>	OBJECT IDENTIFIER ::= {id-at 68}
<i>-- id-at-pkiPath</i>	OBJECT IDENTIFIER ::= {id-at 69}
<i>-- id-at-privPolicy</i>	OBJECT IDENTIFIER ::= {id-at 70}
<i>-- id-at-role</i>	OBJECT IDENTIFIER ::= {id-at 71}
<i>-- id-at-delegationPath</i>	OBJECT IDENTIFIER ::= {id-at 72}

-- Attributs de commande --

id-cat-sequenceMatchType	OBJECT IDENTIFIER ::= {id-cat 1}
id-cat-wordMatchType	OBJECT IDENTIFIER ::= {id-cat 2}
id-cat-characterMatchTypes	OBJECT IDENTIFIER ::= {id-cat 3}
id-cat-selectedContexts	OBJECT IDENTIFIER ::= {id-cat 4}

-- Attributs de notification --

id-not-dSAPProblem	OBJECT IDENTIFIER ::= {id-not 0}
id-not-searchServiceProblem	OBJECT IDENTIFIER ::= {id-not 1}
id-not-serviceType	OBJECT IDENTIFIER ::= {id-not 2}
id-not-attributeTypeList	OBJECT IDENTIFIER ::= {id-not 3}
id-not-matchingRuleList	OBJECT IDENTIFIER ::= {id-not 4}
id-not-filterItem	OBJECT IDENTIFIER ::= {id-not 5}
id-not-attributeCombinations	OBJECT IDENTIFIER ::= {id-not 6}
id-not-contextTypeList	OBJECT IDENTIFIER ::= {id-not 7}
id-not-contextList	OBJECT IDENTIFIER ::= {id-not 8}
id-not-contextCombinations	OBJECT IDENTIFIER ::= {id-not 9}
id-not-hierarchySelectList	OBJECT IDENTIFIER ::= {id-not 10}
id-not-searchControlOptionsList	OBJECT IDENTIFIER ::= {id-not 11}
id-not-serviceControlOptionsList	OBJECT IDENTIFIER ::= {id-not 12}
id-not-multipleMatchingLocalities	OBJECT IDENTIFIER ::= {id-not 13}
id-not-proposedRelaxation	OBJECT IDENTIFIER ::= {id-not 14}
id-not-appliedRelaxation	OBJECT IDENTIFIER ::= {id-not 15}

-- Définitions des problèmes --

id-pr-targetDsaUnavailable	OBJECT IDENTIFIER ::= {id-pr 1}
id-pr-dataSourceUnavailable	OBJECT IDENTIFIER ::= {id-pr 2}
id-pr-unidentifiedOperation	OBJECT IDENTIFIER ::= {id-pr 3}
id-pr-unavailableOperation	OBJECT IDENTIFIER ::= {id-pr 4}
id-pr-searchAttributeViolation	OBJECT IDENTIFIER ::= {id-pr 5}
id-pr-searchAttributeCombinationViolation	OBJECT IDENTIFIER ::= {id-pr 6}
id-pr-searchValueNotAllowed	OBJECT IDENTIFIER ::= {id-pr 7}
id-pr-missingSearchAttribute	OBJECT IDENTIFIER ::= {id-pr 8}
id-pr-searchValueViolation	OBJECT IDENTIFIER ::= {id-pr 9}
id-pr-attributeNegationViolation	OBJECT IDENTIFIER ::= {id-pr 10}
id-pr-searchValueRequired	OBJECT IDENTIFIER ::= {id-pr 11}
id-pr-invalidSearchValue	OBJECT IDENTIFIER ::= {id-pr 12}
id-pr-searchContextViolation	OBJECT IDENTIFIER ::= {id-pr 13}
id-pr-searchContextCombinationViolation	OBJECT IDENTIFIER ::= {id-pr 14}
id-pr-missingSearchContext	OBJECT IDENTIFIER ::= {id-pr 15}
id-pr-searchContextValueViolation	OBJECT IDENTIFIER ::= {id-pr 16}
id-pr-searchContextValueRequired	OBJECT IDENTIFIER ::= {id-pr 17}
id-pr-invalidContextSearchValue	OBJECT IDENTIFIER ::= {id-pr 18}
id-pr-unsupportedMatchingRule	OBJECT IDENTIFIER ::= {id-pr 19}
id-pr-attributeMatchingViolation	OBJECT IDENTIFIER ::= {id-pr 20}
id-pr-unsupportedMatchingUse	OBJECT IDENTIFIER ::= {id-pr 21}

id-pr-matchingUseViolation	OBJECT IDENTIFIER	::=	{id-pr 22}
id-pr-hierarchySelectForbidden	OBJECT IDENTIFIER	::=	{id-pr 23}
id-pr-invalidHierarchySelect	OBJECT IDENTIFIER	::=	{id-pr 24}
id-pr-unavailableHierarchySelect	OBJECT IDENTIFIER	::=	{id-pr 25}
id-pr-invalidSearchControlOptions	OBJECT IDENTIFIER	::=	{id-pr 26}
id-pr-invalidServiceControlOptions	OBJECT IDENTIFIER	::=	{id-pr 27}
id-pr-searchSubsetViolation	OBJECT IDENTIFIER	::=	{id-pr 28}
id-pr-unmatchedKeyAttributes	OBJECT IDENTIFIER	::=	{id-pr 29}
id-pr-ambiguousKeyAttributes	OBJECT IDENTIFIER	::=	{id-pr 30}
id-pr-unavailableRelaxationLevel	OBJECT IDENTIFIER	::=	{id-pr 31}
id-pr-emptyHierarchySelection	OBJECT IDENTIFIER	::=	{id-pr 32}
id-pr-administratorImposedLimit	OBJECT IDENTIFIER	::=	{id-pr 33}
id-pr-permanentRestriction	OBJECT IDENTIFIER	::=	{id-pr 34}
id-pr-temporaryRestriction	OBJECT IDENTIFIER	::=	{id-pr 35}
id-pr-relaxationNotSupported	OBJECT IDENTIFIER	::=	{id-pr 36}

-- Règles de concordance --

-- id-mr-objectIdentifierMatch	OBJECT IDENTIFIER	::=	{id-mr 0}
-- id-mr-distinguishedNameMatch	OBJECT IDENTIFIER	::=	{id-mr 1}
id-mr-caselgnoreMatch	OBJECT IDENTIFIER	::=	{id-mr 2}
id-mr-caselgnoreOrderingMatch	OBJECT IDENTIFIER	::=	{id-mr 3}
id-mr-caselgnoreSubstringsMatch	OBJECT IDENTIFIER	::=	{id-mr 4}
id-mr-caseExactMatch	OBJECT IDENTIFIER	::=	{id-mr 5}
id-mr-caseExactOrderingMatch	OBJECT IDENTIFIER	::=	{id-mr 6}
id-mr-caseExactSubstringsMatch	OBJECT IDENTIFIER	::=	{id-mr 7}
id-mr-numericStringMatch	OBJECT IDENTIFIER	::=	{id-mr 8}
id-mr-numericStringOrderingMatch	OBJECT IDENTIFIER	::=	{id-mr 9}
id-mr-numericStringSubstringsMatch	OBJECT IDENTIFIER	::=	{id-mr 10}
id-mr-caselgnoreListMatch	OBJECT IDENTIFIER	::=	{id-mr 11}
id-mr-caselgnoreListSubstringsMatch	OBJECT IDENTIFIER	::=	{id-mr 12}
id-mr-booleanMatch	OBJECT IDENTIFIER	::=	{id-mr 13}
id-mr-integerMatch	OBJECT IDENTIFIER	::=	{id-mr 14}
id-mr-integerOrderingMatch	OBJECT IDENTIFIER	::=	{id-mr 15}
id-mr-bitStringMatch	OBJECT IDENTIFIER	::=	{id-mr 16}
id-mr-octetStringMatch	OBJECT IDENTIFIER	::=	{id-mr 17}
id-mr-octetStringOrderingMatch	OBJECT IDENTIFIER	::=	{id-mr 18}
id-mr-octetStringSubstringsMatch	OBJECT IDENTIFIER	::=	{id-mr 19}
id-mr-telephoneNumberMatch	OBJECT IDENTIFIER	::=	{id-mr 20}
id-mr-telephoneNumberSubstringsMatch	OBJECT IDENTIFIER	::=	{id-mr 21}
id-mr-presentationAddressMatch	OBJECT IDENTIFIER	::=	{id-mr 22}
id-mr-uniqueMemberMatch	OBJECT IDENTIFIER	::=	{id-mr 23}
id-mr-protocolInformationMatch	OBJECT IDENTIFIER	::=	{id-mr 24}
id-mr-uTCTimeMatch	OBJECT IDENTIFIER	::=	{id-mr 25}
id-mr-uTCTimeOrderingMatch	OBJECT IDENTIFIER	::=	{id-mr 26}
id-mr-generalizedTimeMatch	OBJECT IDENTIFIER	::=	{id-mr 27}
id-mr-generalizedTimeOrderingMatch	OBJECT IDENTIFIER	::=	{id-mr 28}
id-mr-integerFirstComponentMatch	OBJECT IDENTIFIER	::=	{id-mr 29}
id-mr-objectIdentifierFirstComponentMatch	OBJECT IDENTIFIER	::=	{id-mr 30}
id-mr-directoryStringFirstComponentMatch	OBJECT IDENTIFIER	::=	{id-mr 31}
id-mr-wordMatch	OBJECT IDENTIFIER	::=	{id-mr 32}
id-mr-keywordMatch	OBJECT IDENTIFIER	::=	{id-mr 33}
-- id-mr-certificateExactMatch	OBJECT IDENTIFIER	::=	{id-mr 34}
-- id-mr-certificateMatch	OBJECT IDENTIFIER	::=	{id-mr 35}
-- id-mr-certificatePairExactMatch	OBJECT IDENTIFIER	::=	{id-mr 36}
-- id-mr-certificatePairMatch	OBJECT IDENTIFIER	::=	{id-mr 37}
-- id-mr-certificateListExactMatch	OBJECT IDENTIFIER	::=	{id-mr 38}
-- id-mr-certificateListMatch	OBJECT IDENTIFIER	::=	{id-mr 39}
-- id-mr-algorithmIdentifierMatch	OBJECT IDENTIFIER	::=	{id-mr 40}
id-mr-storedPrefixMatch	OBJECT IDENTIFIER	::=	{id-mr 41}
-- id-mr-attributeCertificateMatch	OBJECT IDENTIFIER	::=	{id-mr 42}
-- id-mr-readerAndKeyIDMatch	OBJECT IDENTIFIER	::=	{id-mr 43}
-- id-mr-attributeIntegrityMatch	OBJECT IDENTIFIER	::=	{id-mr 44} <i>obsolète</i>
-- id-mr-attributeCertificateExactMatch	OBJECT IDENTIFIER	::=	{id-mr 45}
-- id-mr-holderIssuerMatch	OBJECT IDENTIFIER	::=	{id-mr 46}
id-mr-systemProposedMatch	OBJECT IDENTIFIER	::=	{id-mr 47}
id-mr-generalWordMatch	OBJECT IDENTIFIER	::=	{id-mr 48}
id-mr-approximateStringMatch	OBJECT IDENTIFIER	::=	{id-mr 49}

ISO/CEI 9594-6:2001 (F)

id-mr-ignoreIfAbsentMatch	OBJECT IDENTIFIER ::= {id-mr 50}
id-mr-nullMatch	OBJECT IDENTIFIER ::= {id-mr 51}
id-mr-zonalMatch	OBJECT IDENTIFIER ::= {id-mr 52}
-- id-mr-authAttIdMatch	OBJECT IDENTIFIER ::= {id-mr 53}
-- id-mr-roleSpecCertIdMatch	OBJECT IDENTIFIER ::= {id-mr 54}
-- id-mr-basicAttConstraintsMatch	OBJECT IDENTIFIER ::= {id-mr 55}
-- id-mr-delegatedNameConstraintsMatch	OBJECT IDENTIFIER ::= {id-mr 56}
-- id-mr-timeSpecMatch	OBJECT IDENTIFIER ::= {id-mr 57}
-- id-mr-attDescriptorMatch	OBJECT IDENTIFIER ::= {id-mr 58}
-- id-mr-acceptableCertPoliciesMatch	OBJECT IDENTIFIER ::= {id-mr 59}
-- id-mr-policyMatch	OBJECT IDENTIFIER ::= {id-mr 60}
-- id-mr-delegationPathMatch	OBJECT IDENTIFIER ::= {id-mr 61}
-- id-mr-pkiPathMatch	OBJECT IDENTIFIER ::= {id-mr 62}
-- contextes --	
id-avc-language	OBJECT IDENTIFIER ::= {id-avc 0}
id-avc-temporal	OBJECT IDENTIFIER ::= {id-avc 1}
id-avc-locale	OBJECT IDENTIFIER ::= {id-avc 2}
-- id-avc-attributeValueSecurityLabelContext	OBJECT IDENTIFIER ::= {id-avc 3}
-- id-avc-attributeValueIntegrityInfoContext	OBJECT IDENTIFIER ::= {id-avc 4}

END -- Types d'attributs sélectionnés

Annexe B

Résumé des types d'attributs

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

La présente annexe résume les types d'attributs sélectionnés qui ont fait l'objet d'une référence ou d'une définition dans la présente Spécification d'annuaire. Elle montre également leur hiérarchie. Les attributs qui partagent une syntaxe ASN.1 commune sont reproduits en retrait au-dessous de cette syntaxe et les attributs qui sont des sous-types d'autres attributs sont en retrait de leurs supertypes. Les attributs collectifs qui sont des sous-types d'un attribut non collectif associé ne sont pas indiqués, mais l'attribut associé est suivi d'un astérisque (*). Les attributs de notification sont désignés par le signe #. Aucune *syntaxe* n'est indiquée pour les attributs dont la spécification de syntaxe fait partie intégrante de la spécification du type d'attribut.

DirectoryString	<i>DistinguishedName</i>
name	distinguishedName
commonName	member
surname	owner
givenName	roleOccupant
initials	seeAlso
generationQualifier	
countryName	<i>FilterItem</i>
localityName *	filterItem #
stateOrProvinceName *	<i>AttributeCombination</i>
organizationName *	attributeCombinations #
organizationalUnitName *	
pseudonym	<i>ContextAssertion</i>
title	contextList #
dmdName	
streetAddress *	<i>ContextCombination</i>
houseIdentifier	contextCombinations #
description	
businessCategory	<i>HierarchySelections</i>
postalCode *	hierarchySelectList #
postOfficeBox *	
physicalDeliveryOfficeName *	<i>SearchControlOptions</i>
knowledgeInformation	searchOptionsList #
PrintableString	<i>ServiceControlOptions</i>
serialNumber	serviceControlOptionsList #
dnQualifier	
destinationIndicator	<i>MultipleMatchingLocalities</i>
telephoneNumber *	multipleMatchingLocalities
NumericString	<i>MRMappings</i>
x121Address *	proposedRelaxation
internationalISDNNumber *	
ObjectIdentifier	<i>Guide</i>
communicationsService	searchGuide
communicationsNetwork	
supportedApplicationContext	<i>EnhancedGuide</i>
dSAPProblem #	enhancedSearchGuide
searchServiceProblem #	
serviceType #	<i>PostalAddress</i>
attributeTypeList #	postalAddress *
matchingRuleList #	registeredAddress
contextTypeList #	
appliedRelaxation #	<i>TelexNumber</i>
Bit String	telexNumber *
uniqueIdentifier	
NameAndOptionalUID	<i>FacsimileTelephoneNumber</i>
uniqueMember	facsimileTelephoneNumber *
	<i>PresentationAddress</i>
	presentationAddress
	<i>ProtocolInformation</i>
	protocolInformation
	preferredDeliveryMethod

Annexe C

Limites supérieures

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

La présente annexe reprend l'ensemble des valeurs proposées pour les limites supérieures attribuées dans les Spécifications d'annuaire; ces valeurs forment le module ASN.1 **UpperBounds**.

UpperBounds {joint-iso-itu-t ds(5) module(1) upperBounds(10) 4}

DEFINITIONS ::=

BEGIN

-- EXPORTER TOUT --

*-- Les types et les valeurs définis dans le présent module sont exportés en vue de l'utilisation dans les autres modules
-- ASN.1 contenus dans les Spécifications d'annuaire et en vue de l'utilisation d'autres applications pour accéder aux
-- services de l'annuaire. D'autres applications peuvent les utiliser pour leur propre compte, mais elles ne doivent pas
-- gêner les extensions et les modifications nécessaires à la mise à jour et à l'amélioration du service d'annuaire.*

ub-answerback	INTEGER ::= 8
ub-business-category	INTEGER ::= 128
ub-common-name	INTEGER ::= 64
ub-content	INTEGER ::= 32768
ub-country-code	INTEGER ::= 4
ub-description	INTEGER ::= 1024
ub-destination-indicator	INTEGER ::= 128
ub-directory-string-first-component-match	INTEGER ::= 32768
ub-domainLocalID	INTEGER ::= 64
ub-international-isdn-number	INTEGER ::= 16
ub-knowledge-information	INTEGER ::= 32768
ub-localeContextSyntax	INTEGER ::= 128
ub-locality-name	INTEGER ::= 128
ub-match	INTEGER ::= 128
ub-name	INTEGER ::= 64
ub-organization-name	INTEGER ::= 64
ub-organizational-unit-name	INTEGER ::= 64
ub-physical-office-name	INTEGER ::= 128
ub-post-office-box	INTEGER ::= 40
ub-postal-code	INTEGER ::= 40
ub-postal-line	INTEGER ::= 6
ub-postal-string	INTEGER ::= 30
ub-privacy-mark-length	INTEGER ::= 128
ub-pseudonym	INTEGER ::= 128
ub-schema	INTEGER ::= 1024
ub-search	INTEGER ::= 32768
ub-serial-number	INTEGER ::= 64
ub-state-name	INTEGER ::= 128
ub-street-address	INTEGER ::= 128
ub-surname	INTEGER ::= 64
ub-tag	INTEGER ::= 64
ub-telephone-number	INTEGER ::= 32
ub-teletex-terminal-id	INTEGER ::= 1024
ub-telex-number	INTEGER ::= 14
ub-title	INTEGER ::= 64
ub-user-password	INTEGER ::= 128
ub-x121-address	INTEGER ::= 15

END -- UpperBounds

Annexe D

Index alphabétique des attributs, des règles de concordance et des contextes

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

La présente annexe fournit la liste alphabétique de tous les attributs et des règles de concordance définis dans la présente Spécification d'annuaire ainsi que le numéro du paragraphe correspondant à la définition.

Applied Relaxation	5.12.16	Integer First Component Match	6.4.1
Approximate String Match	6.6.1	Integer Match	6.2.2
Attribute Combinations	5.12.7	Integer Ordering Match	6.2.3
Attribute Type List	5.12.4	International ISDN Number	5.7.6
Bit String Match	6.2.4	Keyword Match	6.5.2
Boolean Match	6.2.1	Knowledge Information	5.1.1
Business Category	5.5.4	Language Context	7.1
Case Exact Match	6.1.4	Locale Context	7.3
Case Exact Ordering Match	6.1.5	Locality Name	5.3.2
Case Exact Substrings Match	6.1.6	Matching Rule List	5.12.5
Case Ignore List Match	6.1.10	Member	5.10.2
Case Ignore List Substrings Match	6.1.11	Multiple Matching Localities	5.12.14
Case Ignore Match	6.1.1	Name	5.2.1
Case Ignore Ordering Match	6.1.2	Null Match	6.7.2
Case Ignore Substrings Match	6.1.3	Numeric String Match	6.1.7
Common Name	5.2.2	Numeric String Ordering Match	6.1.8
Communications Network	5.7.10	Numeric String Substrings Match	6.1.9
Communications Service	5.7.9	Object Identifier First Component Match	6.4.2
Context Combinations	5.12.10	Octet String Match	6.2.5
Context List	5.12.9	Octet String Ordering Match	6.2.6
Context Type List	5.12.8	Octet String Substrings Match	6.2.7
Country Name	5.3.1	Organizational Unit Name	5.4.2
Description	5.5.1	OrganizationName	5.4.1
Destination Indicator	5.7.8	Owner	5.10.4
Directory String First Component Match	6.4.3	Physical Delivery Office Name	5.6.4
Distinguished Name	5.10.1	Post Office Box	5.6.3
DMD name	5.11.1	Postal Address	5.6.1
DN Qualifier	5.2.8	Postal Code	5.6.2
dSAProblem	5.12.1	Preferred Delivery Method	5.8.1
Enhanced Search Guide	5.5.3	Presentation Address	5.9.1
Facsimile Telephone Number	5.7.4	Presentation Address Match	6.2.10
Filter Item	5.12.6	Proposed Relaxation	5.12.15
General Word Match	6.5.3	Protocol Information	5.9.3
Generation Qualifier	5.2.6	Protocol Information Match	6.2.12
Generalized Time Match	6.3.3	Pseudonym	5.2.10
Generalized Time Ordering Match	6.3.4	Registered Address	5.7.7
Given Name	5.2.4	Role Occupant	5.10.5
Hierarchy Select List	5.12.11	Search Guide	5.5.2
House Identifier	5.3.5	Search Options List	5.12.12
Ignore if Absent Match	6.7.1	Search Service Problem	5.12.1
Initials	5.2.5	See Also	5.10.6

Serial Number	5.2.9	Teletex Terminal Identifier (deleted)	5.7.3
Service Control Options List	5.12.13	Telex Number	5.7.2
Service type	5.12.3	Temporal Context	7.2
State or Province Name	5.3.3	Title	5.4.3
Stored Prefix Match	6.1.12	Unique Identifier	5.2.7
Street Address	5.3.4	Unique Member	5.10.3
Supported Application Context	5.9.2	Unique Member Match	6.2.11
Surname	5.2.3	UTC Time Match	6.3.1
System Proposed Match	6.3.5	UTC Time Ordering Match	6.3.2
Telephone Number	5.7.1	Word Match	6.5.1
Telephone Number Match	6.2.8	X.121 Address	5.7.5
Telephone Number Substrings Match	6.2.9	Zonal Match	6.8

Annexe E

Exemples de règles de correspondance zonale

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

NOTE – Les indications suivantes donnent des exemples relatifs à la définition de la correspondance zonale décrite au § 6.8 de la présente Spécification d'annuaire. Afin de mettre en évidence les situations auxquelles ces exemples sont applicables, le texte définitif de ce paragraphe est conservé, mais reproduit en italiques.

En correspondance zonale, le mécanisme central implémente un mappage à partir de chaînes ou de combinaisons d'assertions, utilisées dans le filtre d'une opération de recherche, vers un ensemble de caractéristiques irréductibles pouvant être possédées par des objets et décrits par des attributs dans les entrées correspondantes. Ce mappage est exprimé sous la forme d'un ensemble d'éléments de filtrage en variante, qui remplacent les éléments de filtrage du filtre original. Les attributs utilisés pour représenter les assertions dans le filtre ne sont pas nécessairement les mêmes que ceux qui sont utilisés pour représenter les caractéristiques de l'objet dans l'entrée. Voici comment une correspondance zonale spécifique peut s'effectuer:

- un utilisateur, qui recherche un abonné téléphonique, un certain M. Smithers vivant à Bracknell, utilise le filtre suivant: `{{localité=Bracknell} AND {nom de famille=Smithers}}`;
- l'annuaire contient une table de mappage géographique (appelée *répertoire toponymique*) qui met en correspondance la ville de Bracknell avec des codes postaux (par exemple RG12 2JL) délimitant des zones dans le district de Bracknell, ce qui a pour effet de convertir le filtre en `{{zone=b1} OR {zone=b2}... } AND {nom=Smithers}`. Ici, b_1, b_2, \dots, b_n sont l'ensemble des codes postaux qui représentent la ville de Bracknell; chaque résidence individuelle possède un seul code postal, alors qu'un grand immeuble ou un lotissement peut en avoir plusieurs. La recherche de correspondance tente de localiser une personne portant le nom de famille indiqué à l'emplacement géographique qui contient une zone commune au code b_1 ou $b_2 \dots$;
- si cette recherche n'aboutit pas, le mappage peut être automatiquement élargi à un plus grand nombre de zones (c'est-à-dire de codes postaux adjacents). Cela permettra peut-être de trouver un abonné dénommé Smithers, vivant dans le village de Newell Green (qui est limitrophe de Bracknell).

Une règle de correspondance par mappage peut tirer parti de variantes nominatives et d'informations redondantes. Elle peut également combiner plusieurs prédicats, par exemple: `{{localité=Newton} AND {localité=Cumbria}}`; elle peut même identifier plusieurs composants dans un même prédicat, par exemple `{localité="Newton, Cumbria"}`. Cet exemple de correspondance peut donc s'appliquer aux cas suivants:

- `{{localité=Bullbrook} AND {nom=Smithers}}`
(Ici, Bullbrook est un district de la commune de Bracknell)
- `{{localité=Bracknell} AND {localité=Bullbrook} AND {nom=Smithers}}`;
- `{{localité=Bullbrook, Bracknell} AND {nom=Smithers}}`
- `{{locality=Berks} AND {locality=Bracknell} AND {locality=Bullbrook} AND {surname=Smithers}}`
(Bracknell s'inscrit dans les limites de l'ancien comté du Berkshire, abrégé en Berks.)
- `{{localité=Berkshire} AND {localité=Bracknell} AND {localité=Bullbrook} AND {nom=Smithers}}`
- `{{localité=East Berks} AND {localité=Bracknell Forest} AND {nom=Smithers}}`
(La nouvelle zone administrative de la région dans laquelle se trouve la commune de Bracknell est appelée "East Berks[hire]"; le district administratif local est dénommé "Bracknell Forest".)
- `{postcode=RG12 2JL} AND {nom=Smithers}}`
(RG12 2JL est l'un des quelque 20 codes postaux de Bullbrook)

Les règles de correspondance zonale sont fondées sur des correspondances géographiques par mappage. Elles utilisent un dictionnaire de noms de localités appelé "répertoire toponymique". Un répertoire toponymique couvrira généralement (c'est-à-dire fournira une base de données géographiques relatives à) un domaine composé d'un seul pays ou d'une seule région. Lors d'une recherche géographique, une interrogation doit être interprétée dans les termes d'un répertoire toponymique spécifique. Un répertoire toponymique met principalement en relation des chaînes toponymiques avec des lieux nommés, identifiés par une ou plusieurs chaînes toponymiques. Exemples de lieux nommés au Royaume-Uni, identifiés par des chaînes toponymiques: "Mogworthy" dans le comté du Devon, "Offleyhoo" dans le Hertfordshire, "Thames Valley" et "London".

Certaines chaînes toponymiques sont mappées directement sur un même lieu nommé mais cela n'est pas toujours possible. Exemples de toponymes ne désignant pas des lieux précis: "Newton" et "Lees", car chacun de ces noms correspond à plusieurs lieux nommés. Un lieu nommé peut donc devoir être identifié par plusieurs toponymes distincts. Par exemple, dans les trois lieux nommés suivants: ("Newton" "Tattenhall" "Cheshire"), ("Newton" "Chester" "Cheshire") et ("Newton" "Cumbria"), les toponymes sont groupés entre les parenthèses.

Un toponyme peut avoir plusieurs composants internes, par exemple "London Heathrow", "Newton Abbott", mais chacun de ces composants est compté comme une seule chaîne soit parce que le nom est incomplet, même localement, s'il n'a pas tous ses composants, soit parce que le nom (par exemple "Abbott") n'est pas sémantiquement un toponyme (aucun toponyme "Abbott" ne figure dans les répertoires toponymiques normaux). Un lieu nommé peut également être identifié par un sous-ensemble de ses noms multiples; par exemple, l'expression ("Newton" "Tattenhall") peut désigner correctement le lieu mentionné plus haut. Dans ce cas cependant, l'expression ("Newton" "Tattenhall" "Cheshire") peut être un groupement plus utile, par analogie avec le lieu "Newtons", qui ne nécessite qu'une précision par comté, par exemple ("Newton" "Cumbria").

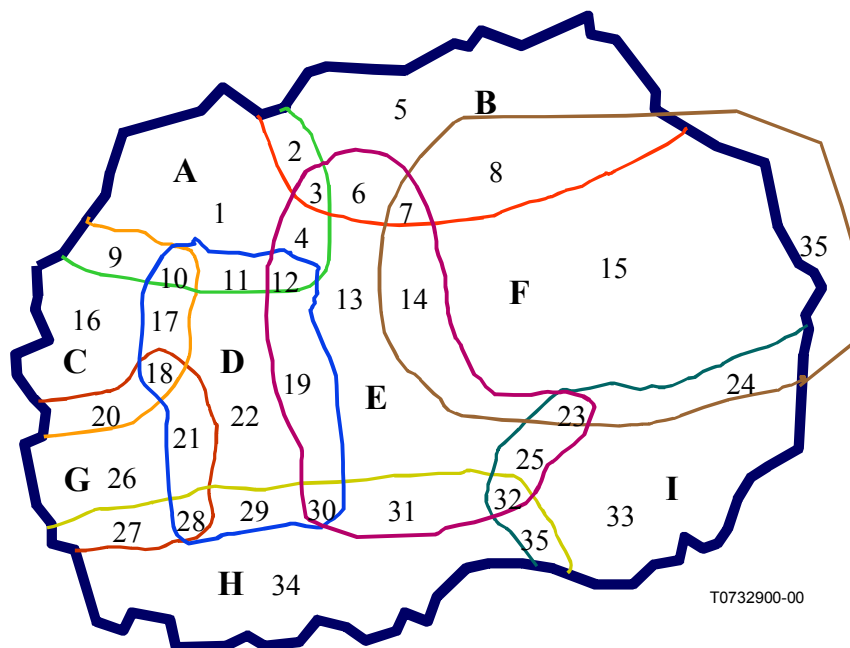


Figure E.1 – Régions, lieux et zones

On trouvera ci-dessous un énoncé plus formel du modèle sous-tendant la correspondance zonale:

- a) *la correspondance zonale est fondée sur l'existence d'un ou de plusieurs répertoires toponymiques qui sont pris en charge à cette fin par des agents DSA. Un répertoire toponymique est un dictionnaire géographique dont le domaine recouvre un pays ou une région toponymique, au moyen d'une base de données appropriée. La sélection du domaine d'une recherche spécifique est effectuée par des moyens locaux. Par exemple, un répertoire toponymique peut couvrir les grandes terres britanniques (Angleterre, Ecosse, Pays de Galles) avec les îles périphériques. Le répertoire toponymique contient des toponymes avec leurs propriétés, y compris des listes de lieux nommés correspondants. Il est pris en charge par des mécanismes d'extraction et de collationnement des propriétés toponymiques indiquées par des attributs combinables de localité. Ce répertoire est tout à fait indépendant de l'arbre DIT. Dans la Figure E.1, la région est délimitée par un trait gras;*
- b) *la région couverte par un répertoire toponymique contient des lieux. Dans la Figure E.1, la limite de région correspond aux zones identifiées par des lettres. Un lieu est une zone géographique nommée distinctement. Les lieux peuvent s'intersecter et peuvent même s'étendre un peu au-delà de la limite d'une région (comme F sur la Figure E.1). Exemples de lieux: Angleterre, Berkshire, Bracknell, Bullbrook (ces quatre lieux s'imbriquant dégressivement) et Thames Valley (qui comprend une partie du Berkshire mais qui en déborde). Les lieux identifiables par référence au répertoire toponymique sont appelés lieux nommés;*

- c) *le répertoire toponymique lui-même est fondé sur des chaînes qui sont des toponymes.* (par exemple "England", Berkshire, "Bracknell", "Bullbrook", "Thames Valley"). *Ceux-ci sont utilisés pour identifier (ou nommer) des lieux nommés. Le nom d'un lieu nommé peut être:*
- *un simple toponyme, comportant éventuellement plusieurs mots* (par exemple "Newton Abbott");
 - *un ensemble de toponymes correspondant généralement à une assez grande surface* (par exemple "Cumbria") *et englobant d'autres toponymes correspondant (dans le contexte) à une plus petite surface* (par exemple "Newton").

En général, un lieu nommé devra être associé, dans le répertoire toponymique, aux noms des lieux l'englobant sur une plus grande échelle, même si ces lieux ne sont pas nécessaires pour une identification unique. Par exemple, le répertoire toponymique devra définir la ville de "Newton Abbott" comme étant accessible aussi bien par "Newton Abbott" que par ("Newton Abbott" "Devon"). Il faudra donc que ce lieu soit associé au toponyme "Devon" (qui se trouve être synonyme de "Devonshire").

Annexe F

Amendements et corrigenda

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

L'édition de la présente Spécification d'annuaire comprend l'amendement suivant:

- Amendement 1 pour l'amélioration de la prise en charge de la Rec. UIT-T F.510 et du codage UTF-8.

L'édition de la présente Spécification d'annuaire comprend les corrigenda techniques qui corrigent les erreurs signalées dans les Rapports ci-après: 211, 237, 238, 241, 246, 251, 253 et 270.

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	Réseaux câblés et 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, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication